



NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

**NUMERICAL MODELING OF ION DYNAMICS IN A
CARBON NANOTUBE FIELD-IONIZED ION THRUSTER**

by

Sarah F. Michael

December 2011

Thesis Advisors:

Dragoslav Grbovic

Marcello Romano

Second Reader:

Oscar Biblarz

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			<i>Form Approved OMB No. 0704-0188</i>	
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.				
1. AGENCY USE ONLY (Leave blank)		2. REPORT DATE December 2011	3. REPORT TYPE AND DATES COVERED Master's Thesis	
4. TITLE AND SUBTITLE Numerical Modeling of Ion Dynamics in a Carbon Nanotube Field-Ionized Thruster			5. FUNDING NUMBERS	
6. AUTHOR(S) Michael, Sarah F.				
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			8. PERFORMING ORGANIZATION REPORT NUMBER	
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			10. SPONSORING/MONITORING AGENCY REPORT NUMBER	
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. Government. IRB Protocol number _____N/A_____.				
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			12b. DISTRIBUTION CODE	
13. ABSTRACT (maximum 200 words) <p>Carbon nanotube field ionization technology has the potential to make ion propulsion feasible for use in micro- and nano-satellites. To better understand the phenomenon and optimize the ion thruster design, it is useful to have an accurate model of the system. Numerical modeling of large-scale electron bombardment ion engines is a relatively mature field, but modeling of field-ionized ion engines is in its infancy. A simpler code may be appropriate for the early modeling stages of carbon nanotube field ionization technology; one such software package is <i>spiffe</i>. <i>Spiffe</i> is intended for modeling axisymmetric radio frequency guns, but it contains all the code necessary for basic modeling of ion optics in a field-ionized ion thruster.</p> <p>This work analyzes the feasibility of <i>spiffe</i> software for use in modeling field-ionized ion thrusters. It also provides detailed procedures for its use. In this work, <i>spiffe</i> is first verified to agree with theoretical predictions of limits in a one-dimensional approximation using electrons. Two primary geometries and their boundary conditions are investigated. The geometry is then varied to test the limits of the one-dimensional approximation. This was further altered to simulate singly-charged Argon ions and verified against theoretical one-dimensional limits.</p> <p>A supplemental user's guide was developed to aid students with minimal programming experience to quickly become familiar with the methods used in <i>spiffe</i> and the impact of program options on the results. A guide to quickly post-processing the data was also developed.</p>				
14. SUBJECT TERMS Field ionization, CNT, numerical modeling, carbon nanotube			15. NUMBER OF PAGES 123	
			16. PRICE CODE	
17. SECURITY CLASSIFICATION OF REPORT Unclassified	18. SECURITY CLASSIFICATION OF THIS PAGE Unclassified	19. SECURITY CLASSIFICATION OF ABSTRACT Unclassified	20. LIMITATION OF ABSTRACT UU	

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

**NUMERICAL MODELING OF ION DYNAMICS IN A CARBON NANOTUBE
FIELD-IONIZED THRUSTER**

Sarah F. Michael
Lieutenant Commander, United States Navy
B.S., University of Michigan, 2000

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN ASTRONAUTICAL ENGINEERING

from the

**NAVAL POSTGRADUATE SCHOOL
December 2011**

Author: Sarah F. Michael

Approved by: Dragoslav Grbovic
Thesis Co-Advisor

Marcello Romano
Thesis Co-Advisor

Oscar Biblarz
Second Reader

Knox T. Millsaps
Chair, Department of Mechanical and Aerospace Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Carbon nanotube field ionization technology has the potential to make ion propulsion feasible for use in micro- and nano-satellites. To better understand the phenomenon and optimize the ion thruster design, it is useful to have an accurate model of the system. Numerical modeling of large-scale electron bombardment ion engines is a relatively mature field, but modeling of field-ionized ion engines is in its infancy. A simpler code may be appropriate for the early modeling stages of carbon nanotube field ionization technology; one such software package is `spiffe`. `Spiffe` is intended for modeling axisymmetric radio frequency guns, but it contains all the code necessary for basic modeling of ion optics in a field-ionized ion thruster.

This work analyzes the feasibility of `spiffe` software for use in modeling field-ionized ion thrusters. It also provides detailed procedures for its use. In this work, `spiffe` is first verified to agree with theoretical predictions of limits in a one-dimensional approximation using electrons. Two primary geometries and their boundary conditions are investigated. The geometry is then varied to test the limits of the one-dimensional approximation. This was further altered to simulate singly-charged Argon ions and verified against theoretical one-dimensional limits.

A supplemental user's guide was developed to aid students with minimal programming experience to quickly become familiar with the methods used in `spiffe` and the impact of program options on the results. A guide to quickly post-processing the data was also developed.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

I.	INTRODUCTION.....	1
A.	BACKGROUND	1
B.	OBJECTIVES	1
C.	ORGANIZATION	2
II.	BACKGROUND	5
A.	ELECTROSTATIC THRUSTERS.....	5
1.	Space Charge	6
2.	Beam Spreading	9
3.	Aspect Ratio.....	9
4.	Perveance	10
5.	Ion Optics.....	10
B.	ION SOURCES	11
1.	Electron Bombardment	11
2.	Field Ionization.....	14
III.	INTRODUCTION TO SPIFFE	17
A.	OVERVIEW	17
B.	PARTICE-IN-CELL (PIC) METHOD.....	17
1.	Particle Mover	18
2.	Particle and Field Weighting	18
C.	SPIFFE METHODS	18
1.	Particle and Field Propagation	19
2.	Space Charge	19
3.	Poisson Correction	20
4.	Lost Particles	20
5.	Boundary Conditions.....	20
6.	Geometry Definition	21
7.	Modifications for Ions.....	21
IV.	METHODOLOGY	23
A.	RESEARCH APPROACH.....	23
1.	Analysis of Simulation Tools.....	23
2.	One-Dimensional Approximation	23
3.	Geometry Variations	24
4.	Ion Variations.....	24
B.	INITIAL CONDITIONS	24
1.	Cathode Placement	24
2.	Initial Momentum	24
3.	Grid Size	24
4.	Time Step	25
5.	Current Density.....	25
6.	Macroparticles.....	25

7.	Simulation Time	26
8.	Boundary Conditions.....	26
9.	Accelerating Voltage.....	26
C.	DATA CAPTURE	26
D.	BASELINE GEOMETRIES	27
1.	Simple Box	27
a.	Boundary Conditions	27
b.	Screens.....	28
2.	Parallel Plate.....	28
a.	Boundary Conditions	28
b.	Screens.....	28
V.	ONE-DIMENSIONAL APPROXIMATION	29
A.	BOX – SPACE CHARGE OFF	29
1.	Current Density.....	29
2.	Electric Field.....	29
3.	Screen Data.....	31
B.	BOX – SPACE CHARGE ON	31
1.	Current Density.....	31
2.	Electric Field.....	32
3.	Screen Data.....	35
C.	PARALLEL PLATE – SPACE CHARGE OFF	35
1.	Current Density.....	35
2.	Electric Field.....	36
3.	Screen Data.....	38
D.	PARALLEL PLATE – SPACE CHARGE ON.....	38
1.	Current Density	38
2.	Electric Field.....	39
3.	Screen Data.....	41
VI.	PARALLEL PLATE VARIATIONS.....	43
A.	OVERVIEW	43
B.	REMOVAL OF ONE PLATE	43
1.	Removal of Left Plate	43
2.	Removal of Right Plate	44
C.	PLATE THICKNESS VARIATIONS	45
1.	Increased Thickness.....	45
2.	Decreased Thickness.....	47
3.	Cavity Size	49
D.	PLATE RADIUS VARIATIONS	49
1.	Decreased Plate Radius	49
2.	Increased Radius.....	50
VII.	SIMULATION TIME VARIATIONS	53
A.	OVERVIEW	53
B.	BOX.....	53
C.	PARALLEL PLATES	54

VIII.	BOX VARIATIONS	55
A.	OVERVIEW.....	55
B.	ASPECT RATIO VARIATIONS	55
1.	AR=200	55
2.	AR=50	56
3.	AR=10	57
4.	AR=5	58
5.	AR=1	59
C.	CONSTANT ASPECT RATIO, GEOMETRY SCALED-UP.....	60
1.	Scale Up $\times 5$	60
2.	Scale Up $\times 10$	61
3.	Scale Up $\times 50$	62
4.	Scale Up $\times 100$	63
IX.	ION SIMULATION.....	65
A.	PROGRAM SET-UP	65
B.	RESULTS	65
X.	ANALYSIS	67
A.	ONE-DIMENSIONAL APPROXIMATION	67
B.	BOX VARIATIONS	68
C.	ION SIMULATION.....	70
XI.	CONCLUSION AND RECOMMENDATIONS.....	71
A.	CONCLUSION	71
B.	FUTURE WORK.....	72
APPENDIX A.	SUPPLEMENTAL USER'S GUIDE	73
A.	USER'S GUIDE	73
B.	SDDS TOOLKIT	73
C.	DOWNLOAD AND INSTALL.....	73
1.	Download Address.	73
2.	Spiffe.....	73
3.	SDDS Toolkit.....	73
4.	Verify Installation	74
D.	PROGRAM STRUCTURE.....	74
E.	GEOMETRY FILE	74
1.	Overview	74
2.	Example – Box Geometry.....	75
F.	RUN FILE.....	75
1.	Overview	75
2.	Define the geometry	76
a.	Cavity	76
b.	Grid.....	76
c.	Geometry Output Files.....	79
d.	Boundary Conditions	81
e.	Radial and Longitudinal Interpolation	81
f.	Example Code	81

3.	Define the Cathode.....	82
a.	Position	83
b.	Current Density	83
c.	Macroparticles.....	83
d.	Start/Stop Time.....	84
e.	Initial Momentum	84
f.	Stiffness	85
g.	Current Density Profile.....	85
h.	Example Code	85
4.	Poisson Correction	86
a.	Correction Interval.....	86
b.	Accuracy	86
c.	Iterations.....	87
d.	Example Code	87
5.	Screens	87
6.	Snapshots	88
7.	Field Output Options.....	89
a.	Field Output.	89
b.	Field Sampling	89
8.	Integration	90
a.	Time Step	90
b.	Finish Time	91
c.	Space Charge.....	91
d.	Sample Code.....	91
G.	POST-PROCESSING.....	92
1.	sddsquery	92
2.	sdds2spreadsheet.....	92
3.	sddscombine.....	92
4.	sddsprocess	93
5.	sddscollapse	94
6.	sddsprintout.....	94
7.	sddscontour.....	94
8.	sddsplot	95
9.	sddssplit.....	95
H.	TROUBLESHOOTING	96
1.	General Troubleshooting.....	96
2.	Files not Found.....	97
APPENDIX B	SCREEN DATA	99
LIST OF REFERENCES.....		103
INITIAL DISTRIBUTION LIST		105

LIST OF FIGURES

Figure 1.	Schematic diagram of an ion thruster. From [1].	5
Figure 2.	One-dimensional ion acceleration for negligible charge density (—) and for space-charge limit (---). From [1].	7
Figure 3.	Comparison of perveance conditions. From [3].	11
Figure 4.	Simplified schematic of an electron bombardment thruster. From [4].	12
Figure 5.	Effect of plasma sheath on initial ion trajectories. From [5].	13
Figure 6.	Illustration of the effect of a tapered screen electrode. From [5].	14
Figure 7.	Schematic of a CNT field-ionized thruster (not to scale). From [7].	15
Figure 8.	Generic order of operations for a PIC scheme. From [11].	17
Figure 9.	Electron beam acceleration in the simple box geometry.	27
Figure 10.	Electron beam acceleration in the parallel plate geometry.	28
Figure 11.	Current density requested vs. delivered, box geometry, space charge off.	29
Figure 12.	E_z at $r=0$ for $J_{req}=300 \text{ kA/m}^2$, space charge off.	30
Figure 13.	Equipotential plot, E_z as a function of z and r , $J_{req}=300 \text{ kA/m}^2$, space charge off.	31
Figure 14.	Current density requested vs. delivered, box geometry, space charge on.	32
Figure 15.	E_z at $r=0$ for $J_{req}=300 \text{ kA/m}^2$, space charge on. Plots are taken at time intervals of 0.05 ns. Figure 16 is provided for scale.	33
Figure 16.	E_z at $r=0$ for $J_{req}=300 \text{ kA/m}^2$, space charge on (snapshot 7 from Figure 15).	34
Figure 17.	Equipotential plots, E_z as a function of z and r , $J_{req}=300 \text{ kA/m}^2$, space charge on. Plots are taken at time intervals of 0.05 ns.	35
Figure 18.	Current density requested vs. delivered, parallel plate, space charge off.	36
Figure 19.	E_z as a function of z at $r=0$. Plots are taken at time intervals of 0.05 ns. Figure 20 is provided for scale.	37
Figure 20.	E_z as a function of z at $r=0$ (snapshot 7 from Figure 19).	37
Figure 21.	E_z as a function of z and r , parallel plate, space charge off.	38
Figure 22.	Current density requested vs. delivered, parallel plate, space charge on.	39
Figure 23.	E_z as a function of z at $r=0$, parallel plate, space charge on. Plots are taken at time intervals of 0.05 ns. Figure 24 is provided for scale.	40
Figure 24.	E_z as a function of z at $r=0$, parallel plate, space charge on (snapshot 7 from Figure 23).	40
Figure 25.	E_z as a function of z and r , parallel plate geometry, space charge on. Plots are taken at time intervals of 0.05 ns.	41
Figure 26.	Left plate removed, electron beam acceleration.	43
Figure 27.	Right plate removed, electron beam acceleration.	44
Figure 28.	Variation of current density with plate thickness.	48
Figure 29.	Variation of current density with plate radius.	51
Figure 30.	Box geometry, $AR=200$.	56
Figure 31.	Box geometry, $AR=50$.	57
Figure 32.	Box geometry, $AR=10$.	58
Figure 33.	Box geometry, $AR=5$.	59
Figure 34.	Box geometry, $AR=1$.	60

Figure 35.	Box geometry, magnification $\times 5$	61
Figure 36.	Box geometry, magnification $\times 10$	62
Figure 37.	Box geometry, magnification $\times 50$	63
Figure 38.	Box geometry, magnification $\times 100$	64
Figure 39.	Ion simulation, box geometry, AR=100.	66
Figure 40.	Variation in current density delivered/theoretical for varying aspect ratios....	69
Figure 41.	Variation in current density delivered/theoretical for varying magnifications, AR=100.	70
Figure 42.	Interleaving grid.....	77
Figure 43.	Geometry interpretation with correct number of grid lines.	80
Figure 44.	Geometry interpretation with incorrect number of grid lines.	80

LIST OF TABLES

Table 1.	Left plate removed, space charge off.....	44
Table 2.	Left plate removed, space charge on.....	44
Table 3.	Right plate removed, space charge off.....	45
Table 4.	Right plate removed, space charge on.	45
Table 5.	Plate thickness 0.015 mm, space charge off.	46
Table 6.	Plate thickness 0.01 mm (original geometry), space charge on.....	46
Table 7.	Plate thickness 0.015 mm, space charge on.....	46
Table 8.	Plate thickness 0.02 mm, space charge on.....	47
Table 9.	Plate thickness 0.005 mm, space charge off.	47
Table 10.	Plate thickness 0.005 mm, space charge on.....	47
Table 11.	Plate thickness 0.005 mm, space charge on, 0.0025 mm longitudinal grid resolution.....	48
Table 12.	Plate thickness 0.001 mm, space charge on, 0.001 mm longitudinal grid resolution.....	48
Table 13.	Wide cavity (0.02 mm margin), original plate thickness, space charge on.	49
Table 14.	Original radius (10 mm), space charge on.	49
Table 15.	Decreased radius (7.5 mm), space charge on.....	50
Table 16.	Decreased radius (7.5 mm), decreased cavity height (8.5 mm).....	50
Table 17.	Increased plate radius (20 mm), space charge on.	50
Table 18.	Increased plate radius (40 mm), space charge on.	51
Table 19.	Increased plate radius (60 mm), space charge on.	51
Table 20.	Varying simulation time, box, space charge off.	53
Table 21.	Varying simulation time, box, space charge on.....	53
Table 22.	Varying simulation time, parallel plate, space charge off.	54
Table 23.	Varying simulation time, parallel plate, space charge on.	54
Table 24.	Theoretical vs. delivered current density, varying AR.	69
Table 25.	Theoretical vs. delivered current density, AR=100, geometry magnified.....	69
Table 26.	Impact of grid size on simulation accuracy.	78
Table 27.	Screen data, box, space charge off.....	99
Table 28.	Screen data, box, space charge on.	100
Table 29.	Screen data, parallel plate, space charge off.....	101
Table 30.	Screen data, parallel plate, space charge on.....	102

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

First and foremost, I would like to thank Professor John Lewellen for his time and patience as I learned how to use the software and navigate its many quirks. Additional thanks go to Michael Borland of Argonne National Laboratory, the creator of the software, for assisting me when I was at a dead end.

I wish to thank my advisors, Dragoslav Grbovic and Oscar Biblarz, for the freedom to work as best suited me.

I thank my husband, Rick, for his understanding and patience as I chase one goal after another. I also thank him for serving his country overseas; keeping me safe so I can do the things I love. For my ally is the Rick. I miss you.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. BACKGROUND

Carbon nanotube field ionization technology has the potential to make ion propulsion feasible for use in micro- and nano-satellites. To better understand the technology and also to optimize the ion thruster design, it is useful to have an accurate model of the entire system. Numerical modeling of standard large-scale electron bombardment ion engines is a relatively mature field, but modeling of field-ionized ion engines is in its infancy. Current numerical modeling schemes are concerned with characteristics of a plasma, the ion source, at the plasma sheath interface. Utilization of an available code may be appropriate for the early modeling stages of carbon nanotube field ionization technology; one such software package is `spiffe`. `Spiffe` is intended for modeling axisymmetric RF guns, but it contains all the code necessary for basic modeling of the ion optics in a field-ionized thruster.

B. OBJECTIVES

Any numerical modeling scheme requires a thorough understanding of the applicability and the limits of the program. This logically begins with a study of the equations and boundary conditions utilized, and a comparison to known or theoretical results. This work analyzes the feasibility of `spiffe` software for use in modeling field-ionized ion thrusters. The research questions to be addressed have been:

- How does `spiffe` differ from current steady-state numerical modeling schemes for ion optics?
- Does `spiffe` accurately model a theoretical one-dimensional situation?
- Can `spiffe` be altered to model positive ions?
- What are the limitations of using `spiffe` to model such ions?

C. ORGANIZATION

Chapter II provides a background on electrostatic thrusters, focusing on relevant equations and theoretical limits. Concepts specific to ion thruster design are introduced.

Chapter III provides an overview of `spiffe`. This section is designed to give the user enough background on `spiffe` to understand the methodology presented in Chapter IV. A more detailed description of `spiffe` is provided as a supplemental user's guide in Appendix A. Chapter III also answers two of the research questions. First, how does `spiffe` differ from current modeling schemes? Second, can `spiffe` be altered to model positive ions?

Chapter IV outlines the research approach and provides a description of initial program set-up. Chapter V presents data from the initial set of simulations. A preliminary analysis of data from Chapter V was used to shape the follow-on simulations, the results of which are presented in Chapters VI and VII.

The first part of the study focused on all commercially available modeling programs and their applicability in modeling carbon nanotube field-ionized thrusters. This is covered in Chapter II. This survey of available modeling programs led to selection of `spiffe` for further study.

The second part of the study focused on familiarization of `spiffe`, program structure, method of coding, equations, and boundary conditions. The result of this study was the development of a supplemental user's guide (Appendix A). This guide was developed to aid students with minimal programming experience in quickly becoming familiar with the methods used in `spiffe` and the impact of program options on the results. A guide to post-processing the data was also developed. Chapter III provides an abbreviated overview of `spiffe` and its methods, as well as the general set-up for the simulations.

Part three used a one-dimensional approximation to verify that `spiffe` agreed with standard results in literature. This part of the study explored various geometries and the impact of boundary conditions. This led to selection of a simplified geometry for further study.

The simplified geometry was varied to test the applicability of the one-dimensional approximation. The code was also altered to simulate singly-charged Argon ions and again verified against theoretical one-dimensional limits.

THIS PAGE INTENTIONALLY LEFT BLANK

II. BACKGROUND

A. ELECTROSTATIC THRUSTERS

Electrical propulsion systems are grouped into three major categories: electrothermal, electrostatic, and electromagnetic. Electrostatic thrusters rely on the Coulomb force to accelerate propellant ions. An ion thruster schematic is shown in Figure 1. This model assumes one-dimensional ion flow in the x -direction. The source electrode is located at $x=0$; the accelerating electrode is located at $x=d$. The potential difference between the source and accelerating electrode is V . The method of ion production is not addressed here; ions are assumed to exist at the source in a uniform distribution. The potential difference between the ion source and the accelerating electrode establishes an electric field. The electric field accelerates the ions from the source through the gaps in the accelerating electrode. The ions are neutralized as they exit the thruster to prevent charge build-up.

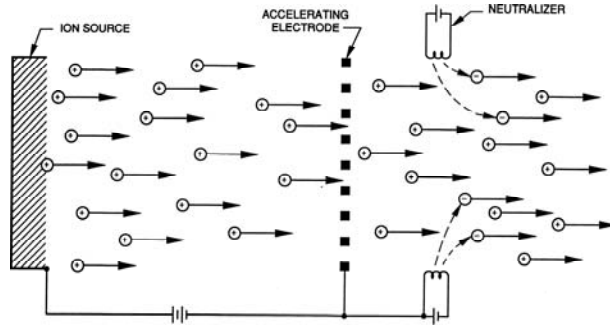


Figure 1. Schematic diagram of an ion thruster. From [1].

The one-dimensional electric field established by the electrodes is given by

$$E = \frac{V}{d} \quad (1)$$

The force F that is experienced by an ion with charge q is

$$F = qE \quad (2)$$

The particle of mass m will be accelerated according to Newton's Second Law.

$$a = \frac{F}{m} \quad (3)$$

Assuming that the particles are emitted with zero velocity at the source, conservation of energy requires that the kinetic energy of the particle equals the electrical energy gained in the field.

$$\frac{1}{2}mv^2 = qV \quad (4)$$

The time required for the particle to traverse the interelectrode space (d) is given by

$$t = \sqrt{\frac{2d}{a}} \quad (5)$$

Combining the above equations, it can be shown that the total thrust generated by the accelerated particles is given by

$$F = I \sqrt{\frac{2Vm}{q}} \quad (6)$$

where I is the current of the ion beam.

1. Space Charge

Equation (6) suggests that the thrust can be increased without bound by simply increasing the current, but this is not possible. In addition to the external electric field established by the electrodes, each ion produces its own electric field. The electric field of the ions counteracts the external field created by the electrodes. This effect will be greatest where the ion density is greatest, at the source electrode. When the electric field from the ions counteracts the external field at the source, no further current may be drawn from the source. This is known as the *space-charge-limited* case and the resultant current density is known as the *saturation current density*. In the case of infinite parallel plates, the saturation current density (J) is given by the Child-Langmuir law.

$$J = \frac{4\varepsilon_0}{9} \sqrt{\frac{2q}{m}} \frac{V^{3/2}}{d^2} \quad (7)$$

where ε_0 is the permittivity of free space, q and m are the charge and mass of the particle, V is the accelerating voltage, and d is the distance between the source and the accelerating electrode.

For a given particle (fixed q/m), the saturation current density will depend only on the accelerating voltage (V) and the interelectrode spacing (d).

Space-charge effects cause the electric field, potential, and ion velocity to vary from the ideal electrostatic case. Figure 2 illustrates the relationship between the ideal case and the space-charge-limited case. In the ideal case (no particles or negligible charge density), the electric field is constant and the potential increases linearly between the electrodes. In the space-charge-limited case, the electric field is nonhomogeneous and approaches zero at the source electrode. The potential distribution between the plates becomes nonlinear.

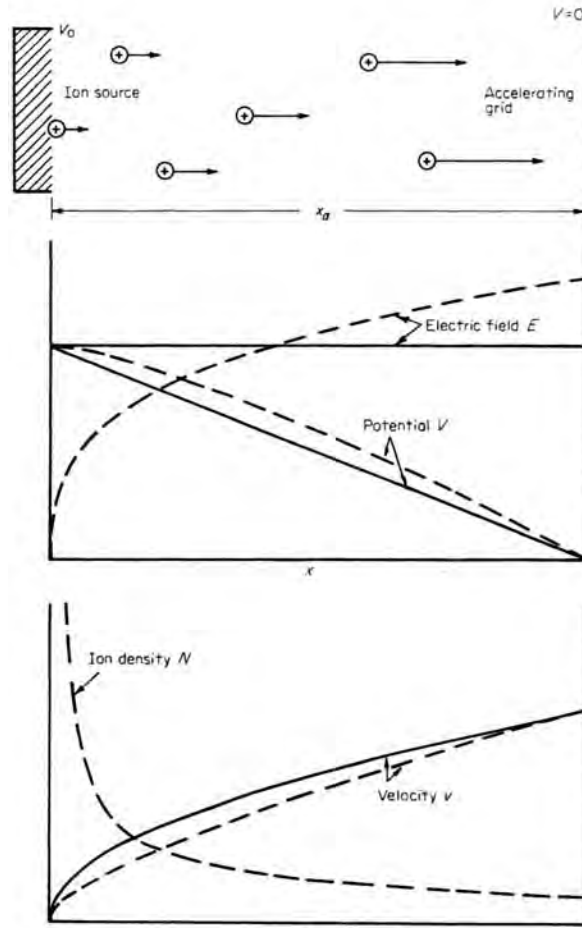


Figure 2. One-dimensional ion acceleration for negligible charge density (—) and for space-charge limit (---). From [1].

The Child-Langmuir law derives from the one-dimensional Poisson equation

$$\frac{d^2V}{dx^2} = -\frac{\rho}{\epsilon_0} \quad (8)$$

where ρ represents the space charge density. A general solution is determined by solving the three-dimensional Poisson equation.

$$\frac{\partial^2 V}{\partial x^2} + \frac{\partial^2 V}{\partial y^2} + \frac{\partial^2 V}{\partial z^2} = -\frac{\rho}{\epsilon_0} \quad (9)$$

Current density is related to space charge density by

$$J = \rho v \quad (10)$$

where velocity (v) is given by Equation (4).

The Poisson equation can be derived from Maxwell's equations. The two relevant equations are Gauss' law

$$\nabla \cdot D = \rho \quad (11)$$

and Faraday's law

$$\nabla \times E = -\frac{\partial B}{\partial t} \quad (12)$$

Assuming a homogeneous, isotropic, linear medium

$$D = \epsilon_0 E \quad (13)$$

Gauss' law becomes

$$\nabla \cdot E = \frac{\rho}{\epsilon_0} \quad (14)$$

In an electrostatic system, the magnetic field is zero. Faraday's law becomes

$$\nabla \times E = 0 \quad (15)$$

Equation (15) indicates that E can be written as the gradient of a scalar (potential).

$$E = -\nabla V \quad (16)$$

Combining equations (14) and (16) results in

$$\nabla \cdot E = -\nabla^2 V = \frac{\rho}{\epsilon_0} \quad (17)$$

which is equivalent to equation (9).

2. Beam Spreading

The electric field from each ion also acts on surrounding ions. The interaction between ions is a repulsive force that causes beam spreading. The electric field (E) from a point charge is given by

$$\bar{E} = \frac{1}{4\pi\epsilon_0} \frac{q}{r^2} \hat{r} \quad (18)$$

where ϵ_0 is the permittivity of free space and r is the radial distance from the point charge to the location of interest.

3. Aspect Ratio

For a circular ion beam of radius r , the current is given by

$$I = J\pi r^2 = J\pi \left(\frac{D^2}{4} \right) \quad (19)$$

Combining this relationship with Equations (6) and (7) gives

$$F = \frac{2\pi\epsilon_0 V^2}{9} \left(\frac{D}{d} \right)^2 \quad (20)$$

The ratio D/d is called the *aspect ratio* (AR). It is a ratio of the effective beam diameter (D) to the interelectrode spacing (d) and uniquely relates the thrust to the square of the applied voltage [1]. Equation (20) suggests that the thrust may achieve large values simply by increasing the aspect ratio. In reality, as D increases, “potential gradients in the radial direction become so great that beam collimation by external electrodes is no longer feasible [2].” A practical limit on the aspect ratio for a single beam accelerated by an external electrode is on the order of unity. Modern ion thrusters have adopted a gridded accelerator structure to overcome this limitation. The ion beam that emerges from each aperture is termed a beamlet. The gridded structure allows each beamlet to approach the practical aspect ratio limit and maximize the thrust produced. The total thrust of the ion beam is a summation of the thrust produced by each beamlet.

4. Perveance

Perveance is a measure of performance of the thruster and is based only on the electric parameters of the system. Perveance is defined as

$$P = \frac{I}{V^{3/2}} \quad (21)$$

and is related to the aspect ratio by

$$P = \frac{\pi\epsilon_0}{9} \sqrt{\frac{2q}{m}} (AR)^2 \quad (22)$$

Perveance is essentially a measure of the current-producing capability of the system. It is useful because it is definable for any accelerator system, whereas aspect ratio is restricted to the plane diode case [1]. Given the perveance, an equivalent plane diode may be developed for more complex systems.

5. Ion Optics

All electrostatic engines consist of an ion source, a means of accelerating those ions, and a means of neutralizing the exhaust. *Ion optics* refers to the accelerating assembly.

In designing the ion optics, it is essential to determine the aspect ratio that results in the optimum perveance for a given beam current and voltage. Alternatively, the design challenge may be to determine the optimum current to voltage ratio for a given aspect ratio. In either case, operating at other than the optimum perveance results in beam impingement on the grids, which both damages the grids and reduces the number of ions available to provide thrust. Figure 3 illustrates under-perveance (a), optimum perveance (b), and over-perveance (c). Over-perveance results in impingement on the accelerating grids; under-perveance results in crossover, overlap of neighboring beamlets, and impingement on downstream surfaces. Optimum perveance results in ideal focusing of the beam without impingement or crossover.

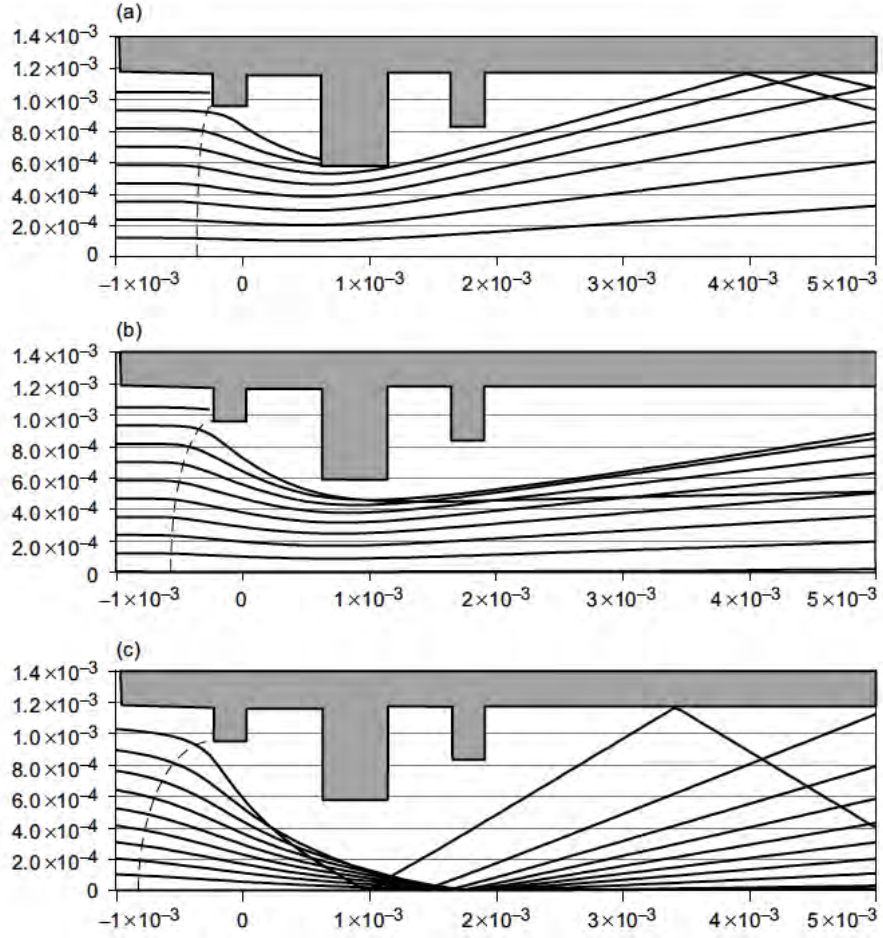


Figure 3. Comparison of perveance conditions. From [3].

B. ION SOURCES

Ions may be produced by several methods: photo-ionization, electron bombardment, field ionization, surface ionization, and thermionic emission. Electron bombardment is a well-established technology [3] [4] and several numerical models have been developed to aid in the design of such thrusters. Field ionization thrusters are under development. Numerical models associated with field ionization typically deal with non-propulsion applications, such as mass spectrometry.

1. Electron Bombardment

In an electron bombardment thruster, ionization is accomplished as electrons

collide with gaseous propellant atoms in the ionization chamber. Ionization of the propellant gas creates both ions and electrons, forming a plasma. A simplified schematic of an electron bombardment thruster is shown in Figure 4. Electrons are attracted to the anode but are forced by the magnetic field to spiral in the chamber, causing numerous collisions and more ionizations. If an electron interacts with an ion, another ionization may occur, thereby producing a doubly-charged ion. Doubly-charged ions have a different charge-to-mass ratio than the singly-charged ions and thus a different trajectory through the ion optics and are considered as losses.

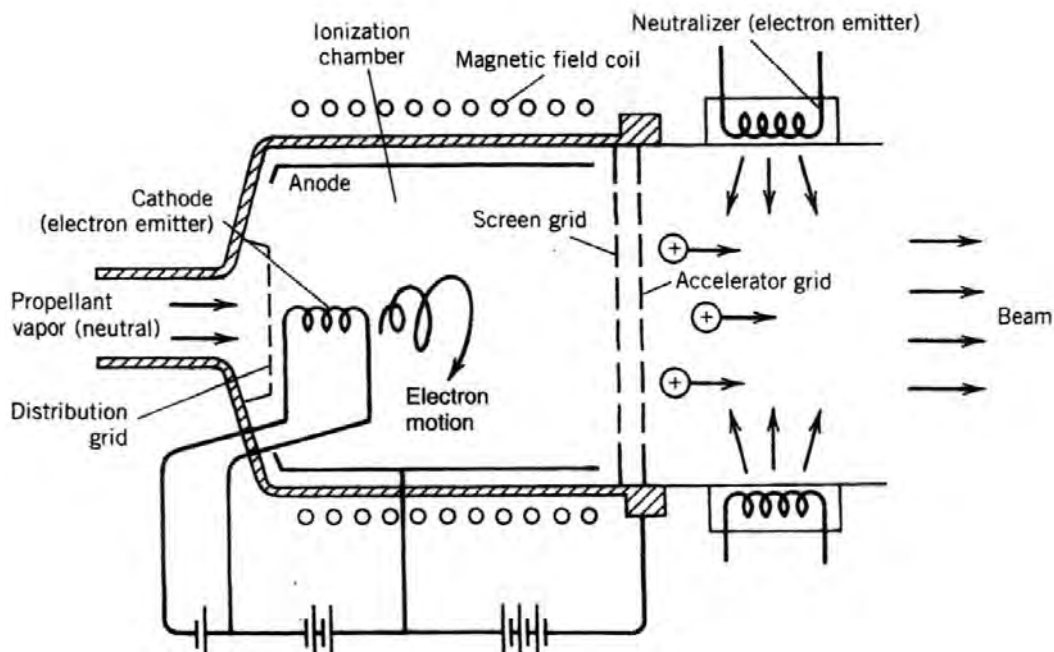


Figure 4. Simplified schematic of an electron bombardment thruster. From [4].

Electrons are removed from the chamber by the radial electric field. The axial electric field moves the ions toward the accelerator grids. Ions must first pass through the screen grid before being accelerated by the accelerating grid. The purpose of the screen grid is to protect the accelerating grid from off-axis ions and provide some confinement of the electrons in the plasma [3].

The presence of plasma in these engines has a significant impact on the design of the ion optics. Figure 5 illustrates the various ways in which the shape of the plasma sheath can serve to shape the ion beam. Each case represents a change in the accelerating

voltage. When there is no accelerating voltage applied (a), the plasma extends into the accelerating region and forms a sheath at each electrode. As the voltage is increased, the plasma sheath is pushed back toward the discharge chamber (b) and the plasma boundary takes on a convex shape (c). As the voltage is increased, the plasma boundary recedes further into the discharge chamber and takes on a concave shape (d). The concave shape is desirable because it provides initial beam shaping.

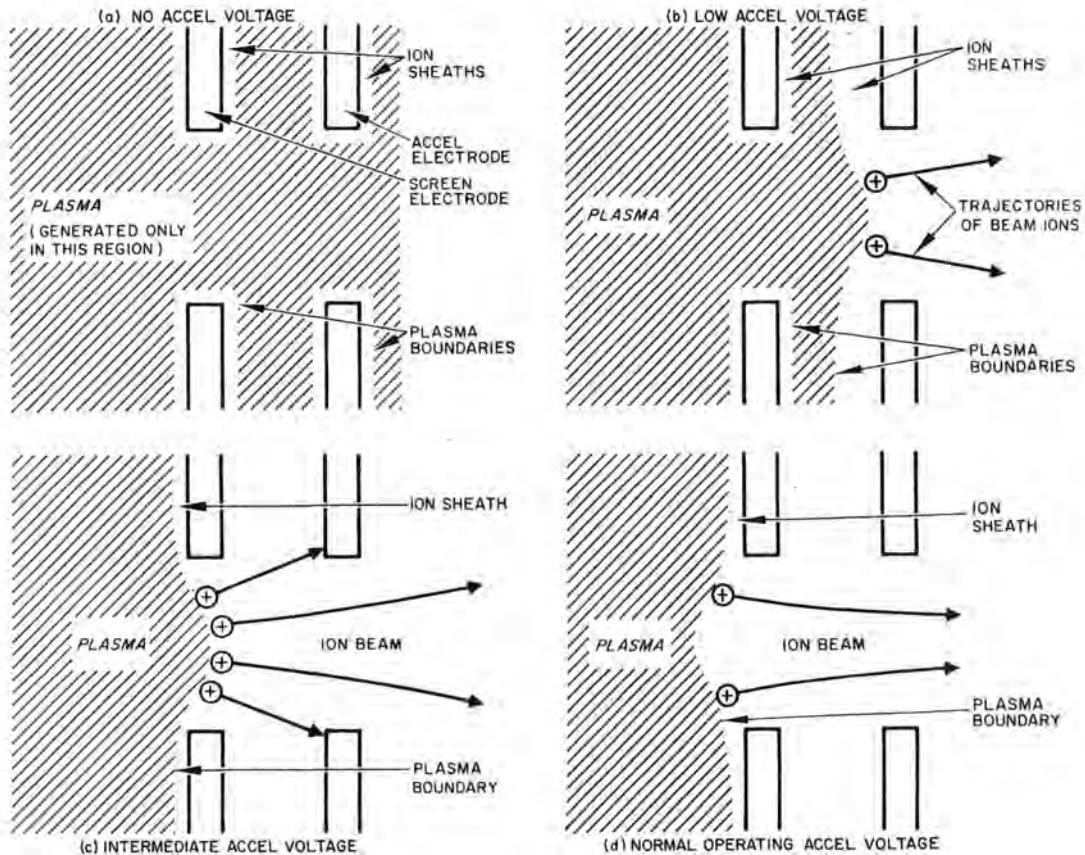


Figure 5. Effect of plasma sheath on initial ion trajectories. From [5].

While the shape of the plasma sheath is not directly at the disposal of the designer, the ion trajectories and source shape are determined by the accelerating voltage and current density. The variation in the shape of the plasma sheath due to changing operating conditions must be considered in the electrode design. Electron bombardment thrusters utilize a tapered screen aperture so that the ion current and trajectories are as

insensitive as possible to the shape of the plasma sheath [6]. Figure 6 illustrates variations in the plasma sheath and the benefit of a tapered screen electrode for different operating conditions.

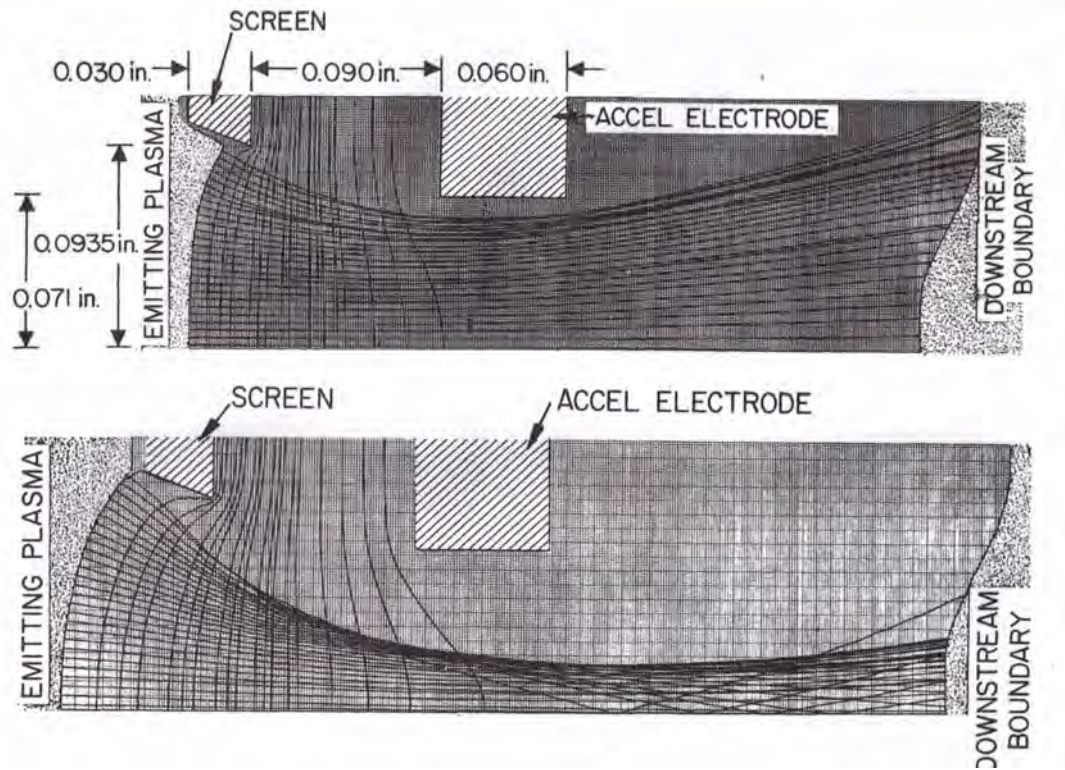


Figure 6. Illustration of the effect of a tapered screen electrode. From [5].

2. Field Ionization

Field ionization occurs when an electron is stripped from a particle by a strong electric field. It is essentially field emission in reverse, and many parallels can be drawn between field ionization and surface ionization [6]. A schematic of a carbon nanotube (CNT) field-ionized thruster is shown in Figure 7. Propellant flows through a porous surface and past the CNTs. The external electric field is amplified at the tips of the CNTs, becoming strong enough to cause the valence electron to tunnel from a propellant atom into a CNT. The electron is absorbed by the metallic surface, leaving only the propellant ion to flow through the engine to provide thrust. This is different from a plasma source, from which electrons and neutral particles may escape to impinge on the

ion optics. A planar arrangement of CNTs makes it appropriate to model the ion source as planar, with no concern for the shape of a plasma sheath.

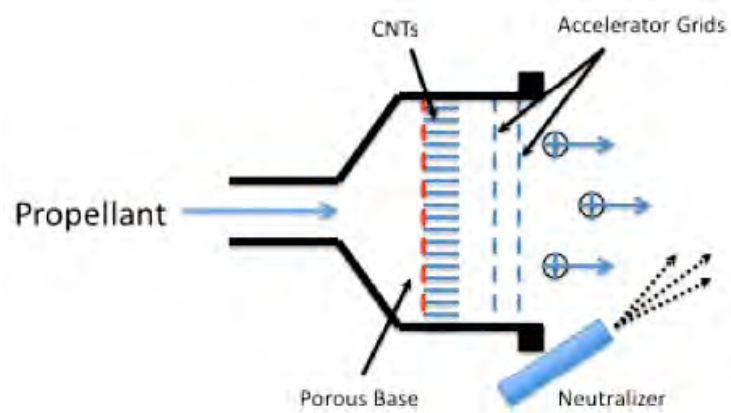


Figure 7. Schematic of a CNT field-ionized thruster (not to scale). From [7].

Field ionization is superior to electron bombardment in that it produces only singly-charged ions [6]. This allows the ion optics to be optimized for a given charge to mass ratio. In an electron bombardment thruster, a doubly-charged ion will follow a different trajectory and may impinge on the ion optics. Numerical simulations must account for the doubly-charged ions when determining particle trajectories. This is not a concern with field ionization sources. The field ionization chamber designer need only be concerned with maximizing the *propellant mass utilization factor*, which measures the number of ions that can be delivered to the accelerator [8].

THIS PAGE INTENTIONALLY LEFT BLANK

III. INTRODUCTION TO SPIFFE

A. OVERVIEW

Spiffe (*S*pace charge and *I*ntegration *F*orces *F*or *E*lectrons) is a 2.5-D fully electromagnetic particle-in-cell (PIC) code for radiofrequency electron gun (RF gun) simulation and similar systems with cylindrical symmetry. The program is inherently unsteady and uses Maxwell's equations and the Lorentz force to compute the time-evolution of fields and particles in a metal cavity. It has been used at NPS to study transmission of intense electron beams through apertures [9] and suppression of current fluctuations in an intense electron beam [10]. *Spiffe* is open source software written in C programming language.

B. PARTICLE-IN-CELL (PIC) METHOD

PIC codes assume that one simulation particle consists of many physical particles. The simulation particle is termed a *macroparticle* or *superparticle*. It exists only as a point charge; it has no volume. The computer code solves equations of motion and calculates macroquantities (e.g., density, current density) from the position and velocity of the macroparticles. The charge-to-mass ratio of a macroparticle is the same as the charge to mass ratio of the physical particle, which means that the macroparticle follows the same trajectory as the physical particle. The generic scheme of a PIC simulation is illustrated in Figure 8.

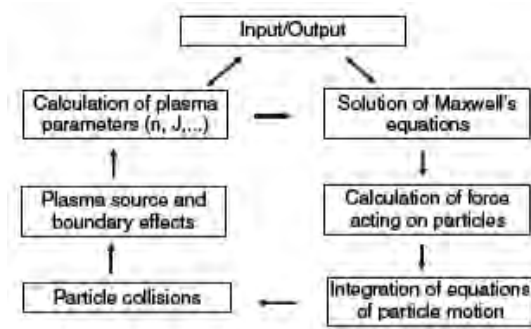


Figure 8. Generic order of operations for a PIC scheme. From [11].

1. Particle Mover

The portion of the code that solves the equations of motion and follows the trajectory of each macroparticle is termed the particle mover. One method of coding the particle mover, and the method employed in `spiffe`, is the *leap-frog method*. The leap-frog method is an explicit solver, meaning that it depends on forces from the previous time step. Explicit solvers are simpler and faster than implicit solvers, but their stability requires a smaller time step [11]. The simulation is divided into discrete time moments (Δt) and the physical quantities (A) are calculated according to:

$$\begin{aligned} t &\rightarrow t_k = t_0 + k\Delta t \\ A(t) &\rightarrow A_k = A(t = t_k) \end{aligned} \tag{23}$$

where $k=0,1,2,\dots$, t is the time, t_0 the initial moment, and A denotes any physical quantity. The leap-frog method calculates some of the physical quantities (B) between time steps.

$$\begin{aligned} t_{k+1/2} &= t_0 + (k+1/2)\Delta t \\ B(t) &\rightarrow B_{k+1/2} = B(t = t_{k+1/2}) \end{aligned} \tag{24}$$

For example, the particle position may be calculated at each time step and the particle velocity between time steps. In this example, particle position is calculated using the velocity from the previous half-step. The particle position is updated and the new particle velocity is calculated from the new position.

2. Particle and Field Weighting

Particle weighting and field weighting are terms used to describe the calculation of macroquantities and field data inside the simulation region. The simulation space is divided into a grid and the desired data is calculated at grid points or interpolated between grid points.

C. SPIFFE METHODS

A robust description of the equations and methods used in `spiffe` is provided in [12]. This section is intended as a brief overview of the basic methods used in `spiffe`.

1. Particle and Field Propagation

This section summarizes the methods described in [12]. In `spiffe`, the particle mover uses position (z, r) and momentum (p_z, p_r) . The position data is calculated at each time step $(t=t_0+n\Delta t)$ and advanced to the next time step $(t=t_0+(n+1)\Delta t)$ according to

$$\begin{aligned} z &\rightarrow z + \frac{p_z c}{\gamma} \Delta t \\ r &\rightarrow r + \frac{p_r c}{\gamma} \Delta t \end{aligned} \tag{25}$$

where c is the speed of light, γ is the Lorentz factor, and $p=\beta\gamma$. Momentum is given in terms of electron mass and the speed of light ($m_e c$).

The electric field is calculated at each time step; the magnetic field is calculated between time steps. At each time step, the electric fields are calculated using the magnetic field and current values from the previous half-step. The magnetic fields are then advanced to the next half-step using the just-calculated electric fields.

Particle-induced currents are needed to calculate the space-charge effect on the electric field, and these currents are computed from particle velocities. Particle position is stepped with the electric field and particle momentum is stepped with the magnetic field (between time steps).

$$\begin{aligned} p_z &\rightarrow p_z + \Delta t \frac{Q}{Mc} \left(E_z + \frac{p_r c}{\gamma} B_\phi \right) \\ p_r &\rightarrow p_r + \Delta t \frac{Q}{Mc} \left(E_r + \frac{p_z c}{\gamma} B_\phi \right) \end{aligned} \tag{26}$$

2. Space Charge

Particle-induced fields (i.e., space charge) are calculated using equivalent current density. Equivalent current density is calculated from the charges, momenta, and positions of the macroparticles. `Spiffe` locates the four grid points that surround the particle and allocates the charge between the upper and lower pairs using linear weighting. The current density is calculated from the charge and velocity of the

macroparticle and the volume of the cell surrounding the grid point. The resultant current density is split between the left and right pairs using linear weighting [12].

`Spiffe` allows the user to run simulations with or without space charge included. With the space charge off, the simulation provides results that are consistent with basic electromagnetics.

3. Poisson Correction

The field propagation routine uses two of Maxwell's equations: Ampere's law and Faraday's law. It does not use the divergence equation. Accumulated errors in the field propagation routine lead to non-zero divergence. `Spiffe` corrects for this by solving the Poisson equation at user-defined intervals during the simulation and clearing any fictitious charge build-up.

4. Lost Particles

`Spiffe` is strictly an axi-symmetric code. Particles that impact a metal surface or outer boundary are lost. Particles that are lost to the centerline ($r=0$) are reflected back into the simulation region.

5. Boundary Conditions

Reference [13] indicates that specification of the potential on a closed surface defines a Dirichlet boundary condition, while specification of the electric field on the surface defines a Neumann boundary condition. In `spiffe`, the user is only able to specify the potential of boundaries, which implies that Dirichlet is the only possible boundary condition. However, [12] uses Maxwell's equations and properties of even and odd functions to show how both Neumann and Dirichlet boundary conditions are implemented in `spiffe`. When a Neumann boundary condition is specified in `spiffe`, the electric field is perpendicular to the boundary. Conversely, the Dirichlet boundary condition requires that the electric field be parallel to the boundary.

6. Geometry Definition

The user defines the geometry using a series of points, which are connected by lines or arcs. Surfaces inside the cavity are assumed to be metal; there is no provision for changing the type of material. The source electrode may be placed anywhere; it does not need to correspond to physical boundary.

7. Modifications for Ions

Spiffe provides an option for increasing the mass of the particle to simulate ions. This is accomplished through use of the “stiffness” option. There is no provision for changing the charge of each particle. This implies that only singly-charged ions can be simulated.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. METHODOLOGY

A. RESEARCH APPROACH

1. Analysis of Simulation Tools

`Spiffe` was installed on a Windows-based laptop computer. The first part of the study involved a brief investigation of various electrode configurations to select which of them was most suitable for the problem at hand.

2. One-Dimensional Approximation

In order to test the simulation performance, a simple one-dimensional scenario was created to verify basic electrostatics (space charge off) and agreement with the Child-Langmuir law (space charge on). With the space charge off, the delivered current density should match the requested current density. This verifies that the code is working properly and that post-processing of the data was done correctly. With the space charge on, the delivered current density should be limited to the saturation current density given by the Child-Langmuir law.

In order to approximate a one-dimensional situation, a parallel plate geometry with an aspect ratio of 100 was created. The cathode radius was based on anticipated modeling of the setup described in [8].

The theoretical saturation current density is calculated from the perveance, which is determined using the aspect ratio. First, the perveance is calculated using Equation (22). Then the current is calculated using Equation (21). The current is converted to current density using the cathode radius.

As `spiffe` was designed for electrons, all simulations were done with electrons as the particle being accelerated. Chapter IX investigates the possibility of using `spiffe` to simulate ions.

3. Geometry Variations

After the model was verified to agree with theoretical predictions, the geometry was varied to test the extent of the applicability of theoretical predictions. First, the aspect ratio was varied by increasing and decreasing the interelectrode spacing. Next, the aspect ratio was maintained at 100 while the geometry was scaled up.

4. Ion Variations

In order to investigate the ability of `spiffe` to model ion flow, the simple box geometry was again utilized. The initial set of simulations (AR=100, varying current density) was repeated for both space charge on and off.

B. INITIAL CONDITIONS

A `spiffe` run has several variables for the user to select, including grid size, cathode placement, initial momentum, simulation time. These can be overwhelming when trying to select the “right” initial conditions for a simulation. The following initial conditions were selected arbitrarily.

1. Cathode Placement

In all simulations, the physical cathode was drawn at $z=0$. Particles were released slightly to the right of the cathode.

2. Initial Momentum

`Spiffe` allows the user to specify an initial momentum of the particles when they are released from the cathode. Initial momentum was set to zero for all simulations.

3. Grid Size

Proper grid sizing is essential to balancing accuracy and reasonable simulation time. A mesh that is too coarse will provide quick but inaccurate results. A mesh that is too fine will be prohibitive in terms of CPU memory and simulation time. The grid lines

were selected to provide a resolution approximately one to two orders of magnitude less than the smallest dimension of geometry. This resolution provided the best balance of accuracy and reasonable run time. Grid sizing is addressed in detail in Appendix A.

4. Time Step

Time step is directly related to grid size. If the number of grid lines doubles, the time step required for stability is halved. As the size of the time step decreases, `spiffe` must perform more calculations. This results in a longer run time, which can exceed twelve hours in some cases. The grid size, simulation time, and time step must be chosen carefully to balance accuracy with run time.

`Spiffe` provides feedback on the minimum time step required for stability. In all simulations, a time step was selected to meet the stability requirement.

5. Current Density

`Spiffe` allows the user to specify a constant current density or to load a current density profile (e.g., half-gaussian ramp-up). A constant current density was used for all simulations. Current density was the primary variable in the simulations.

This report frequently compares current density *requested* versus current density *delivered*. The requested current density is the value that is specified by the user in the `spiffe` code. The current density delivered is the output from the simulation. It is calculated by recording the total charge to pass through a screen over a given time period.

6. Macroparticles

`Spiffe` allows the user to specify macroparticles per step or electrons per macroparticle. These options are discussed further in Appendix A. In all simulations, eight macroparticles per step was used.

7. Simulation Time

Simulation time refers to the time specified in the `spiffe` code; run time refers to the actual time that the simulation took to finish. Selection of the simulation time is critical for ensuring enough time has passed to allow the simulation to reach steady-state, but not so long as to lead to excessive run times.

Equation (5) provides the flight time for a particle. The simulation time was selected by first determining the flight time of a particle. The flight time was increased by a factor of 15 and rounded up slightly to give the simulation time. This method provided a reasonable balance of accuracy and run time.

8. Boundary Conditions

The default boundary conditions were used for all boundaries. These apply only to the cavity boundaries and not to any of the surfaces inside the cavity. The upper, left, and right boundaries are Neumann; the lower is Dirichlet.

- Upper: Neumann, $E_z = 0$ at $r = r_{max}$
- Left: Neumann, $E_r = 0$ at $z = z_{min}$
- Right: Neumann, $E_r = 0$ at $z = z_{max}$
- Lower (centerline): Dirichlet, $E_r = 0$ at $r = 0$

9. Accelerating Voltage

Early simulations used an arbitrary voltage that resulted in electric field strength of 1 MV/m. As geometries were varied, the voltage was adjusted to maintain the electric field strength at 1 MV/m.

C. DATA CAPTURE

The primary method of extracting data from `spiffe` was the use of diagnostic screens. Screens capture location, charge, momentum, and position data for every

macroparticle in the beam. Three screens were placed in the beam path to capture particle data, one near the cathode, one near the anode, and one in the middle of the gap. The screens extend from $r=0$ to $r=r_{max}$. The current density delivered was calculated from the data obtained from the screens. Plots included in this report use the current density from the middle screen.

D. BASELINE GEOMETRIES

1. Simple Box

The simplest geometry is a box that is 0.1 mm wide (z) and 10 mm tall (r). The cathode is located at $0 < r < 5$ mm.

$$AR = \frac{D}{d} = \frac{2(5 \text{ mm})}{0.1 \text{ mm}} = 100 \quad (27)$$

When the simulation starts, electrons are accelerated from the cathode (0 V) to the anode (100 V). The cathode and anode are simply the left and right borders of the box, as shown in Figure 9.

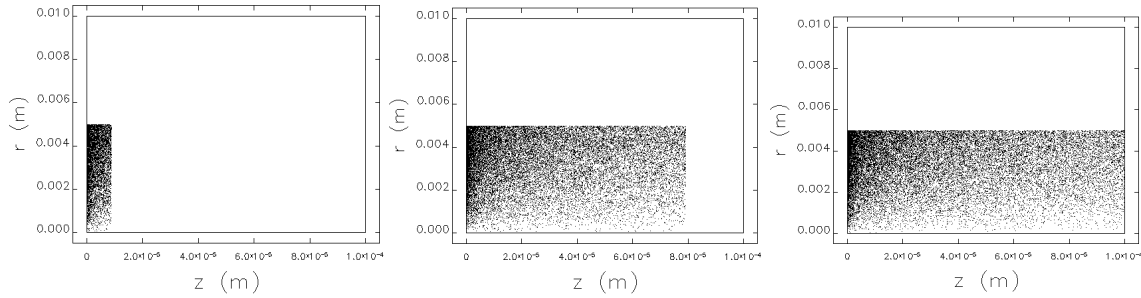


Figure 9. Electron beam acceleration in the simple box geometry.

a. Boundary Conditions

The left and right boundaries may be set to either Neumann or Dirichlet, but the upper is limited to Neumann and the lower (centerline) is limited to Dirichlet. In the simple box geometry, the left and right boundary conditions must be set to Neumann. If they were set to Dirichlet, then E_z would be zero at the left and right boundaries, and there would be no field to accelerate the electrons across the gap.

b. Screens

Three screens are placed in the beam path to capture particle data. The screens are located at 0.01 mm, 0.05 mm, and 0.09 mm.

2. Parallel Plate

The parallel plate geometry uses the same interelectrode spacing and cathode radius as the simple box. The plates are 0.01 mm thick and are separated by 0.1 mm. The cathode is assigned a potential of 0 V; the anode 100 V. The geometry and beam are shown in Figure 10.

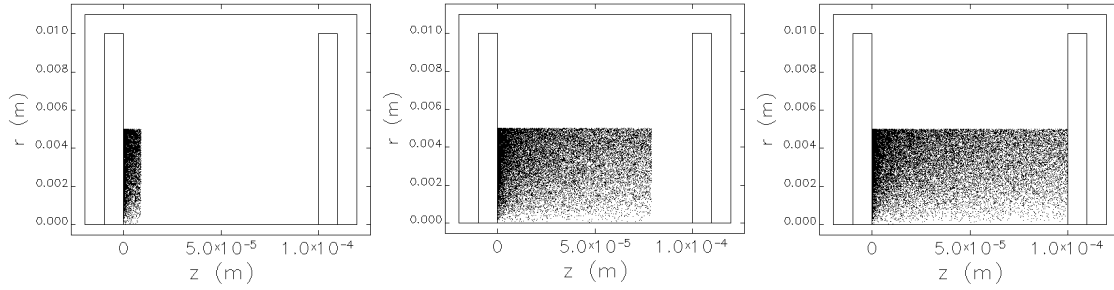


Figure 10. Electron beam acceleration in the parallel plate geometry.

a. Boundary Conditions

The boundary conditions apply to the borders of the cavity. As with the box geometry, they are set to their default values. Any boundaries drawn inside the simulation region are assumed to be metal surfaces. As such, the transverse electric field at the surface of the plates is zero.

b. Screens

The screens are placed at the same location as in the box geometry: 0.01 mm, 0.05 mm, and 0.09 mm.

V. ONE-DIMENSIONAL APPROXIMATION

A. BOX – SPACE CHARGE OFF

1. Current Density

With the space charge off, `spiffe` returns the requested current density (Figure 11). For example, if the code specified 200 kA/m^2 , `spiffe` delivered approximately 200 kA/m^2 . This verifies that the code is working properly and that post-processing of the data was done correctly.

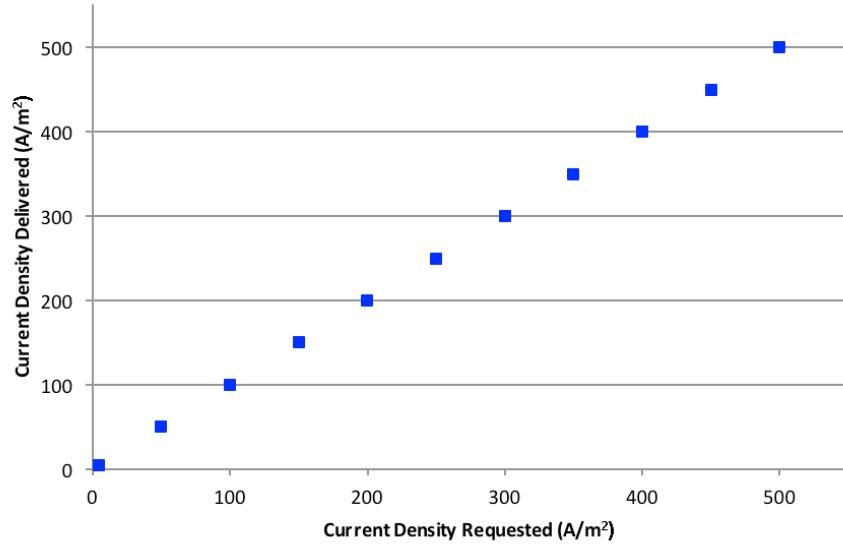


Figure 11. Current density requested vs. delivered, box geometry, space charge off.

2. Electric Field

With the space charge off, the electric field between the cathode and anode should be constant.

$$E_z = \frac{100 \text{ V}}{0.1 \text{ mm}} = 1 \times 10^6 \text{ V/m} \quad (28)$$

The electric field at the centerline ($r=0$) as a function of z is shown in Figure 12.

Each plot represents a time snapshot at intervals of 0.05 ns and all are plotted to the same scale. With the space charge off, the magnitude of the electric field remains constant at 1 MV/m.

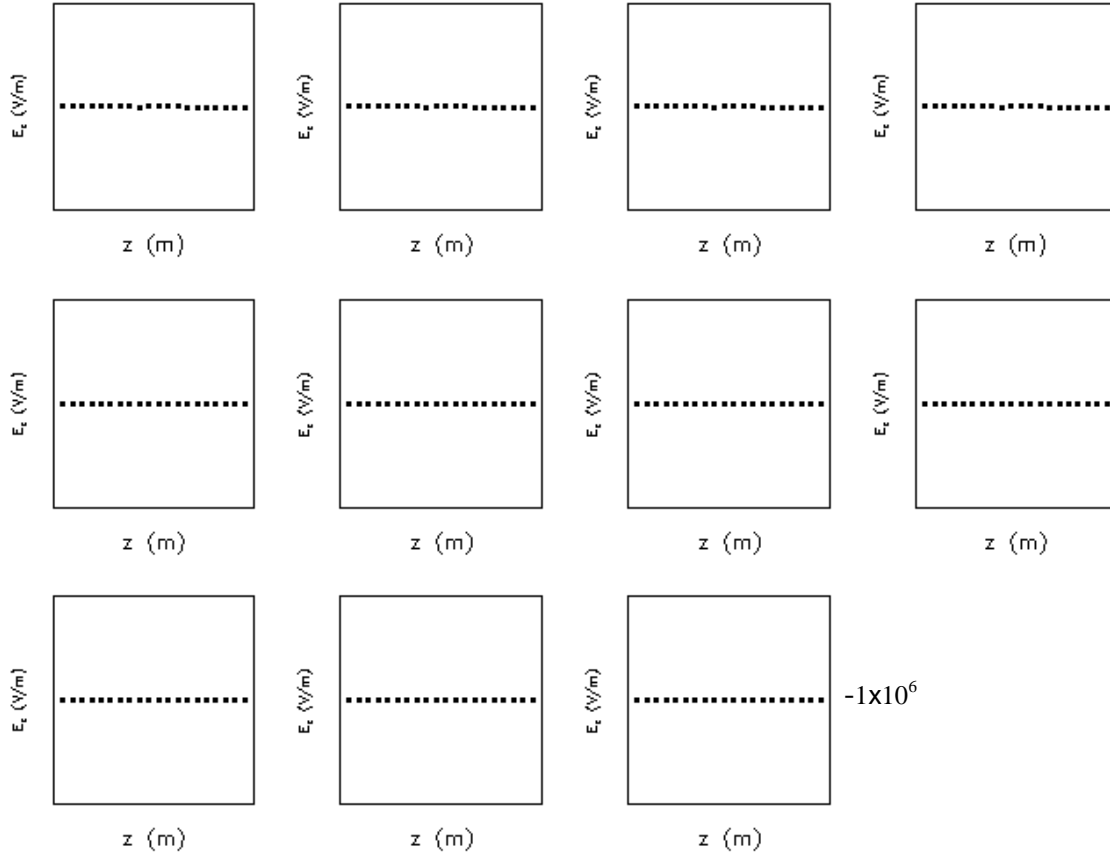


Figure 12. E_z at $r=0$ for $J_{\text{req}}=300 \text{ kA/m}^2$, space charge off.

The plot of the equipotential lines, Figure 13, indicates a constant field for all z and r .

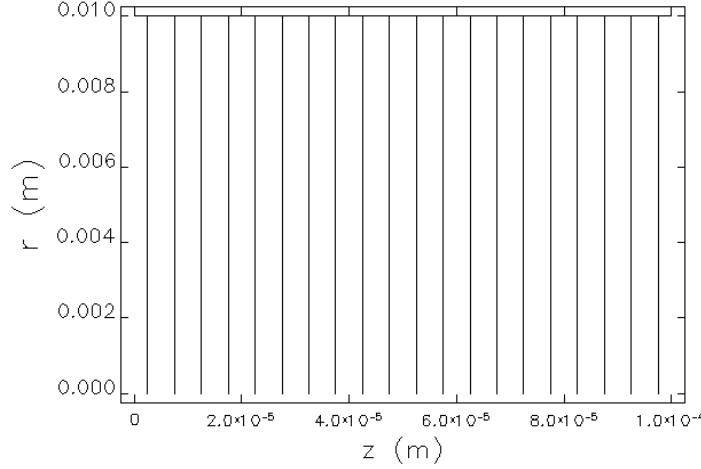


Figure 13. Equipotential plot, E_z as a function of z and r , $J_{\text{req}}=300 \text{ kA/m}^2$, space charge off.

3. Screen Data

Data from the screens is contained in Appendix B. This data shows that the average longitudinal and radial momentums do not vary with requested current density. The maximum radial coordinate of the particles does not exceed 5 mm and is constant for all values of requested current density.

B. BOX – SPACE CHARGE ON

1. Current Density

With the space charge on, the simulation should be limited by saturation current density. The expected saturation current density is calculated using Equations (12) and (13).

$$P = \frac{\pi \epsilon_0}{9} \sqrt{\frac{2q}{m}} (AR)^2 = \frac{\pi (8.85 \times 10^{-12})}{9} \sqrt{\frac{2(1.602 \times 10^{-19})}{9.109 \times 10^{-31}}} (100)^2 = 0.01832 \quad (29)$$

$$I = PV^{3/2} = (0.01832)(100)^{3/2} = 18.32 \quad (30)$$

$$J = \frac{I}{A} = \frac{18.32 \text{ A}}{\pi (5 \text{ mm})^2} = 233 \frac{\text{kA}}{\text{m}^2} \quad (31)$$

Figure 14 shows that current density for the described configuration, simulated in spiffe, levels off at a slightly greater than expected saturation current density, approximately 270 kA/m^2 .

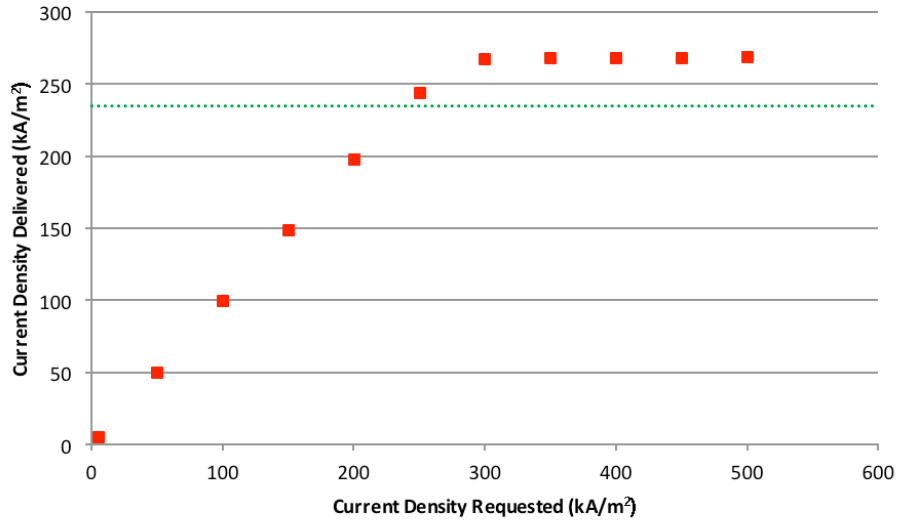


Figure 14. Current density requested vs. delivered, box geometry, space charge on.

2. Electric Field

Electric field snapshots are shown in Figure 15. Figure 16 is a magnified plot of the seventh plot. Each plot represents a time snapshot at intervals of 0.05 ns ; the first plot shows the electric field before particles are released. The requested current density is 300 kA/m^2 . The expectation is that the electric field will be zero at the cathode, as illustrated in the generic case shown in Figure 2. The electric field does not go to zero at the cathode, but it decreases significantly and the field takes on the general shape of the expected distribution.

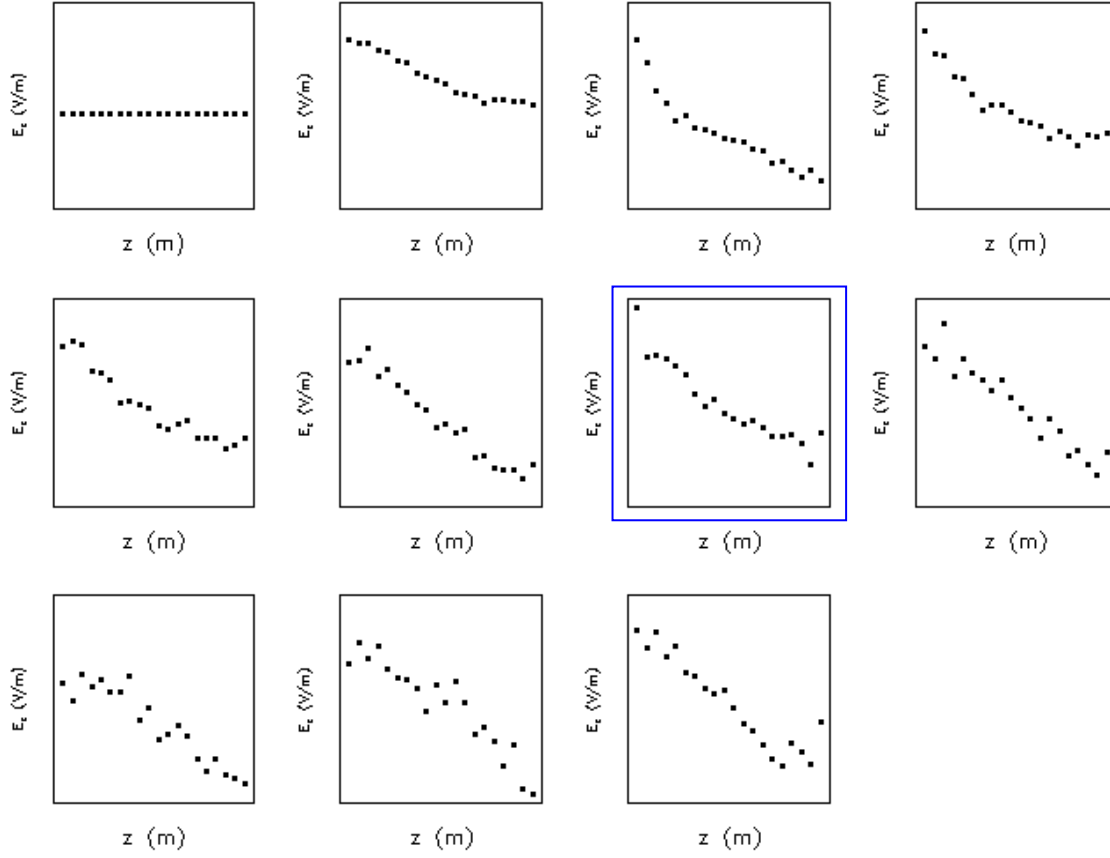


Figure 15. E_z at $r=0$ for $J_{\text{req}}=300 \text{ kA/m}^2$, space charge on. Plots are taken at time intervals of 0.05 ns. Figure 16 is provided for scale.

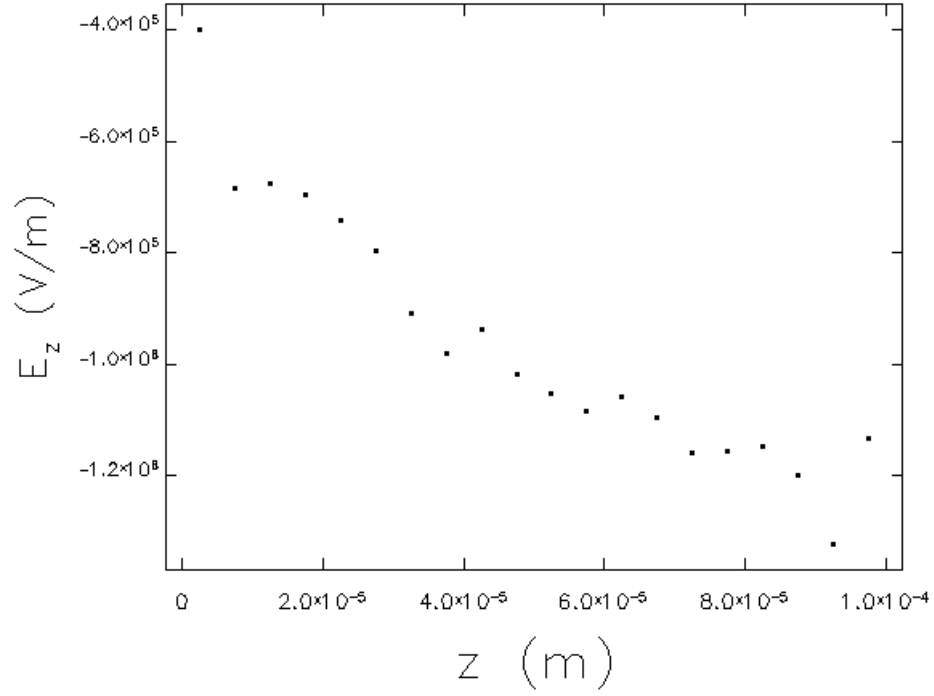


Figure 16. E_z at $r=0$ for $J_{\text{req}}=300 \text{ kA/m}^2$, space charge on (snapshot 7 from Figure 15).

Electric field contour plots are shown in Figure 17. These show significant interaction between the electric field of the particles and that of the electrodes.

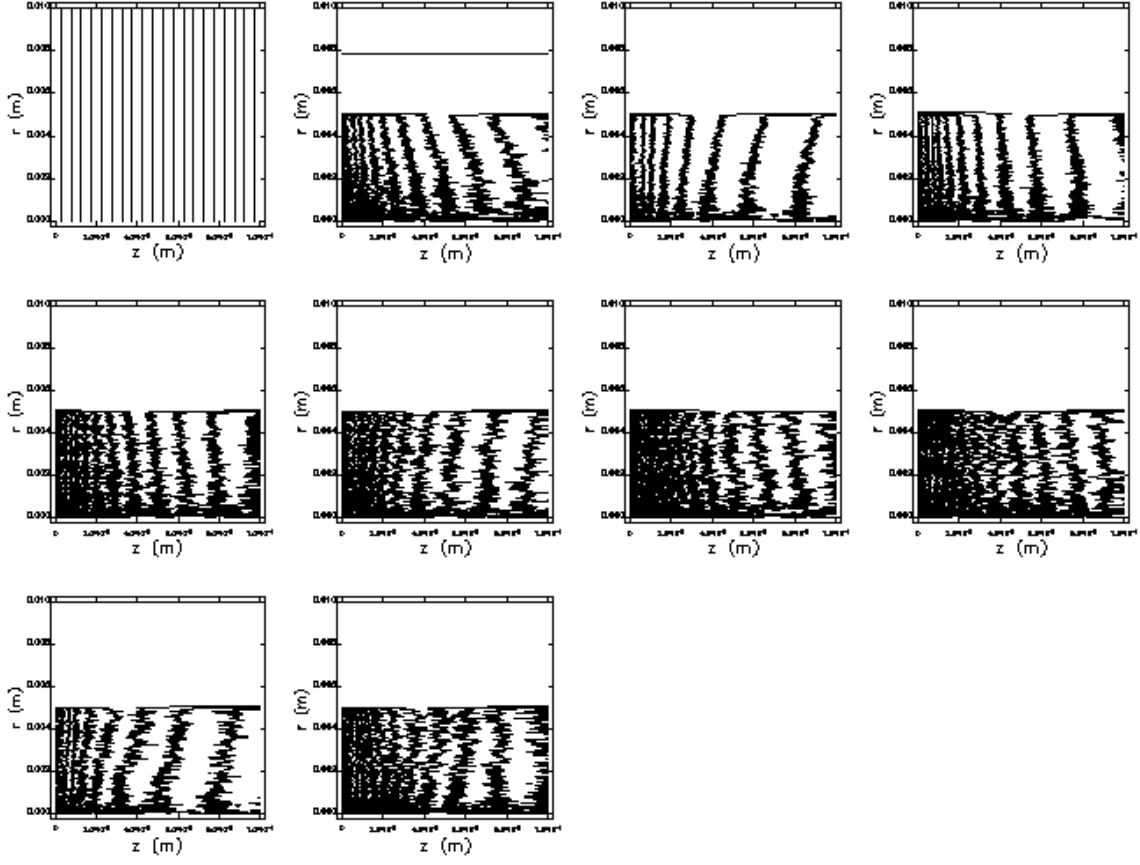


Figure 17. Equipotential plots, E_z as a function of z and r , $J_{\text{req}}=300 \text{ kA/m}^2$, space charge on. Plots are taken at time intervals of 0.05 ns.

3. Screen Data

Data from the screens is contained in Appendix B. With the space charge on, the average longitudinal and radial momentums vary with requested current density. The maximum radial coordinate of the particles exceeds 5 mm, indicating beam spreading.

C. PARALLEL PLATE – SPACE CHARGE OFF

1. Current Density

With the space charge off, `spiffe` returns the requested current density (Figure 18). For example, if the code specified 200 kA/m^2 , `spiffe` delivered approximately

200 kA/m². This verifies that the code is working properly and that post-processing of the data was done correctly. This also provides a comparison to the box geometry.

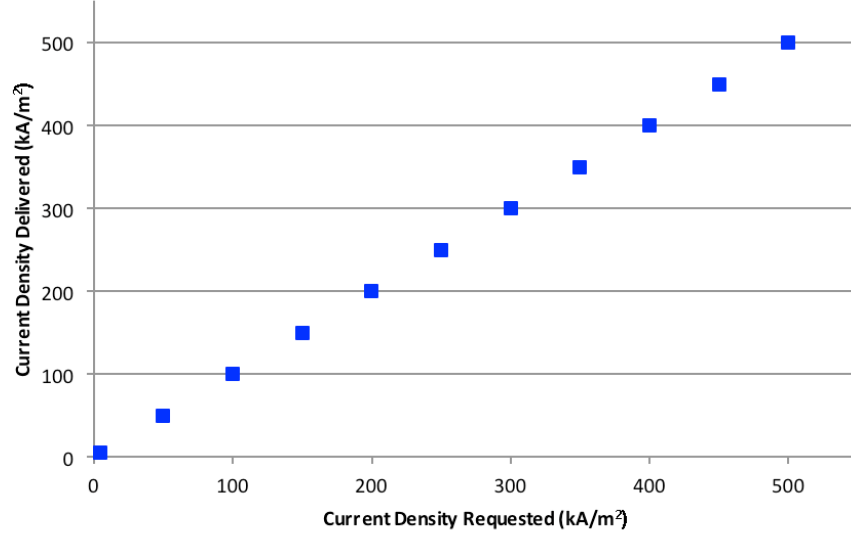


Figure 18. Current density requested vs. delivered, parallel plate, space charge off.

2. Electric Field

The electric field as a function of z at $r=0$ is shown in Figure 21. The electric field at the centerline is constant ($E_z=1$ MV/m) for all points between the plates, with the exception of the two points closest to the electrodes ($E_z=0$). These points did not appear in the box geometry. These points are located on the E_z grid, which does not correspond to the grid that defines the physical boundaries (see Appendix A).

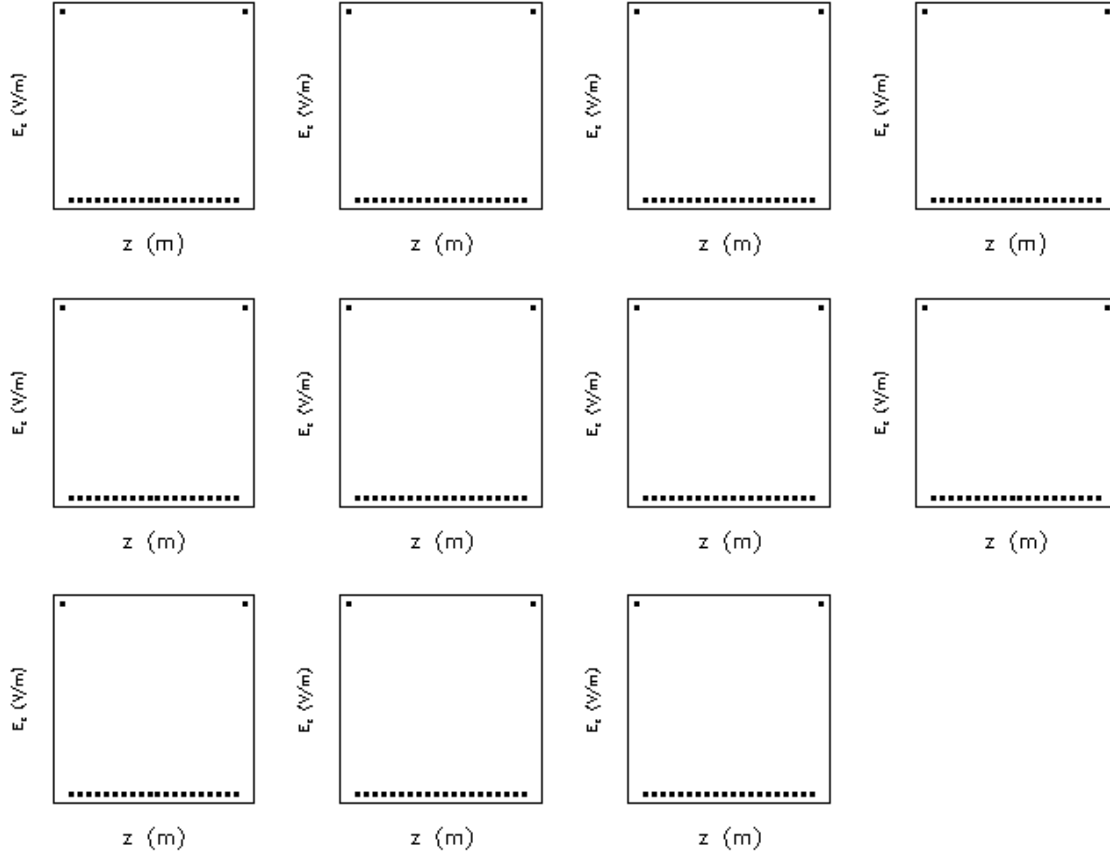


Figure 19. E_z as a function of z at $r=0$. Plots are taken at time intervals of 0.05 ns.

Figure 20 is provided for scale.

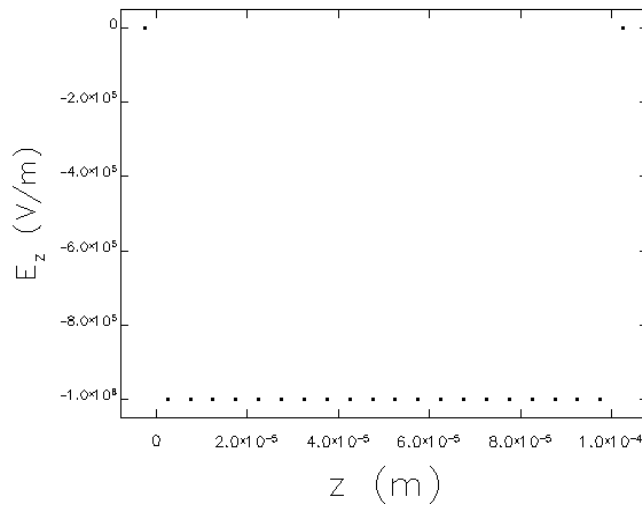


Figure 20. E_z as a function of z at $r=0$ (snapshot 7 from Figure 19).

The electric field contour plot is shown in Figure 21. (This plot only shows the contours from $z=0$ mm to $z=0.1$ mm and $r=0$ mm to $r=10$ mm in order to match the contour plot from the box geometry.) This plot indicates a difference from the box geometry. Each plot uses 20 contour lines, but the parallel plates do not show the regularly spaced equipotential lines seen in Figure 13.

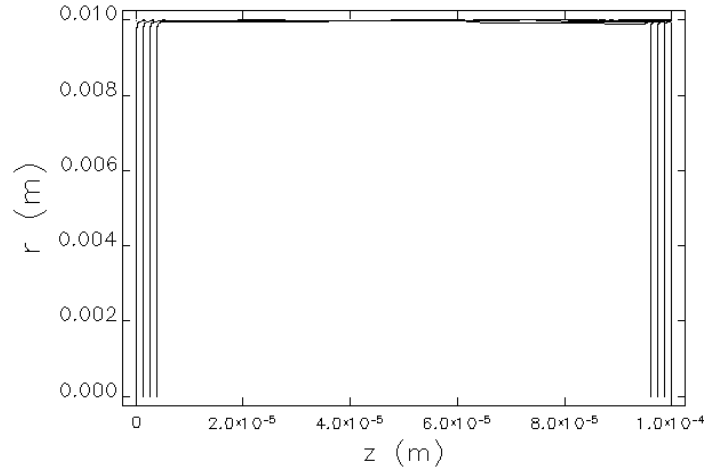


Figure 21. E_z as a function of z and r , parallel plate, space charge off.

3. Screen Data

Data from the screens is contained in Appendix B. Despite the differences in the electric field, the screens return the same data as the box geometry.

D. PARALLEL PLATE – SPACE CHARGE ON

1. Current Density

The expected saturation current density is the same as that calculated in Equation (23). Figure 22 shows that `spiffe` levels off at a saturation current density less than expected, approximately 190 kA/m².

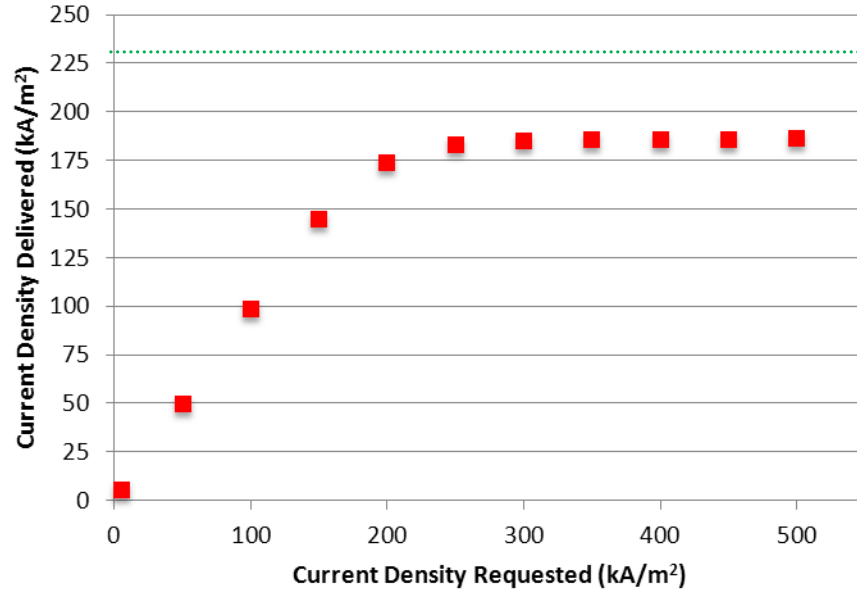


Figure 22. Current density requested vs. delivered, parallel plate, space charge on.

2. Electric Field

The electric field at the centerline ($r=0$) as a function of z is shown in Figure 23. Each plot represents a time snapshot at intervals of 0.05 ns; the first plot shows the electric field before particles are released. The requested current density is 300 kA/m². The electric field again takes on the distribution from Figure 2, with the exception of the two points near the electrodes. These points remain at zero, as they were when the space charge was off (Figure 19). Figure 24 shows a zoomed plot of the seventh plot of Figure 23.

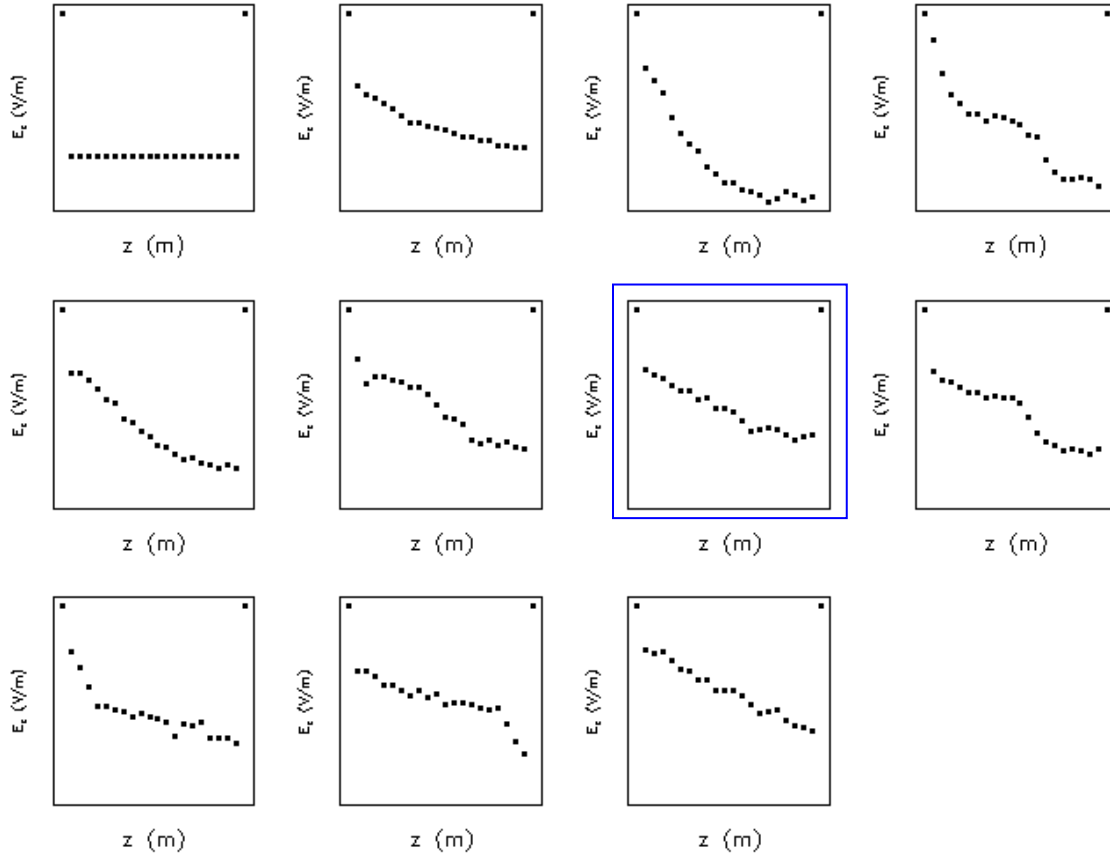


Figure 23. E_z as a function of z at $r=0$, parallel plate, space charge on. Plots are taken at time intervals of 0.05 ns. Figure 24 is provided for scale.

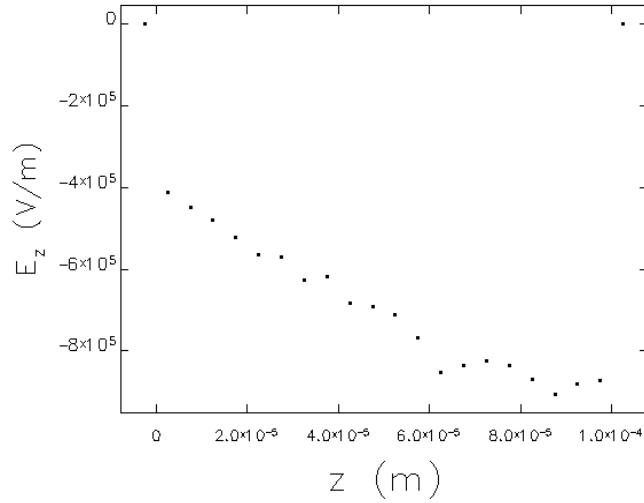


Figure 24. E_z as a function of z at $r=0$, parallel plate, space charge on (snapshot 7 from Figure 23).

The equipotential plots (Figure 25) show differences from the box geometry. In general, the equipotential lines from the box geometry are more evenly spaced.

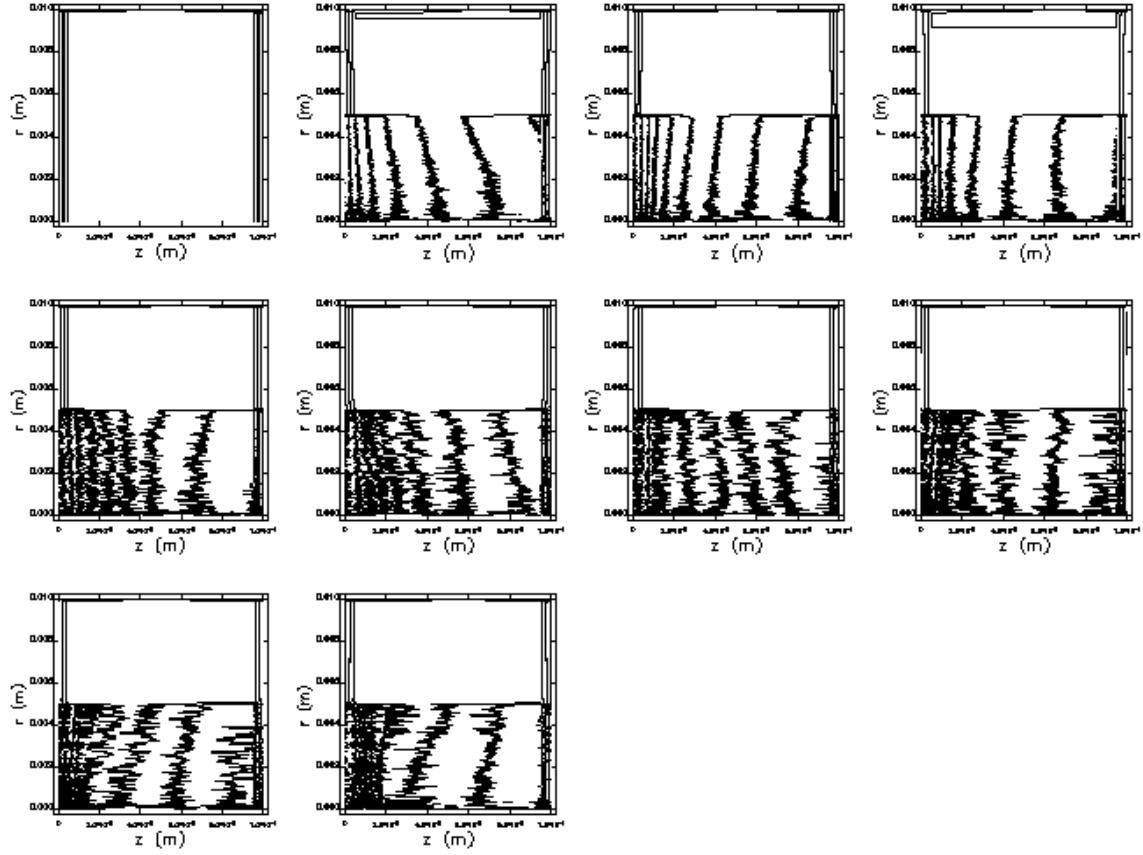


Figure 25. E_z as a function of z and r , parallel plate geometry, space charge on. Plots are taken at time intervals of 0.05 ns.

3. Screen Data

Data from the screens is contained in Appendix B. This data provides insight regarding the differences between the box and parallel plate geometries.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. PARALLEL PLATE VARIATIONS

A. OVERVIEW

The parallel plate and box geometries were expected to provide similar, if not the same, results. Since this was not the case, several additional geometry variations were investigated. Each variation used a requested current density of 300 kA/m^2 . This value was selected because it is above the saturation current density. Values below the saturation current density have minimal impact on the results.

B. REMOVAL OF ONE PLATE

Removal of either the left or right plate results in a saturation current density that falls between the saturation current densities calculated in the case of the box geometry and the full parallel plate geometry.

1. Removal of Left Plate

When the left plate is removed, the left boundary becomes the cathode. The boundary condition at $z=0$ is Neumann ($E_r=0$). The gap is still 0.1 mm. The potential of the cavity boundary is set to 0 V with the default boundary conditions. The remaining plate is set to 100 V and the potential of the cavity boundaries is set to 0 V.

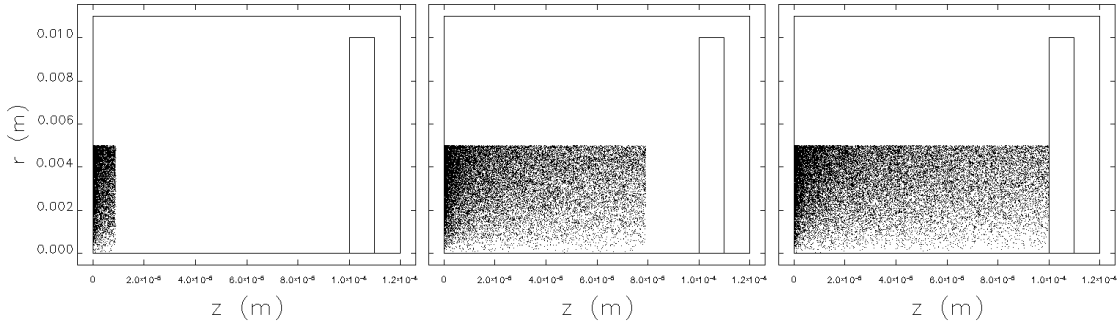


Figure 26. Left plate removed, electron beam acceleration.

With the space charge off, the results (Table 1) are consistent with the parallel plate and box geometries.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
300	0	299.4907	6.256E-03	0	4.999995
	1	299.4482	1.399E-02	0	4.999995
	2	299.5006	1.877E-02	0	4.999995

Table 1. Left plate removed, space charge off.

With the space charge on, the current delivered falls between the box geometry and full parallel plate geometry. The results are listed in Table 2.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
300	0	218.2687	3.239E-03	3.299E-06	5.001634
	1	215.8934	1.132E-02	1.688E-05	5.011986
	2	215.3945	1.717E-02	1.626E-05	5.021632

Table 2. Left plate removed, space charge on.

2. Removal of Right Plate

When the right plate is removed, the cathode is the left plate and the right boundary becomes the anode (Figure 27). The boundary condition at $z=z_{\text{max}}=0.1$ mm is Neumann ($E_r=0$). The potential at the left, top, and bottom boundaries is 0 V; the right boundary is set to 100 V.

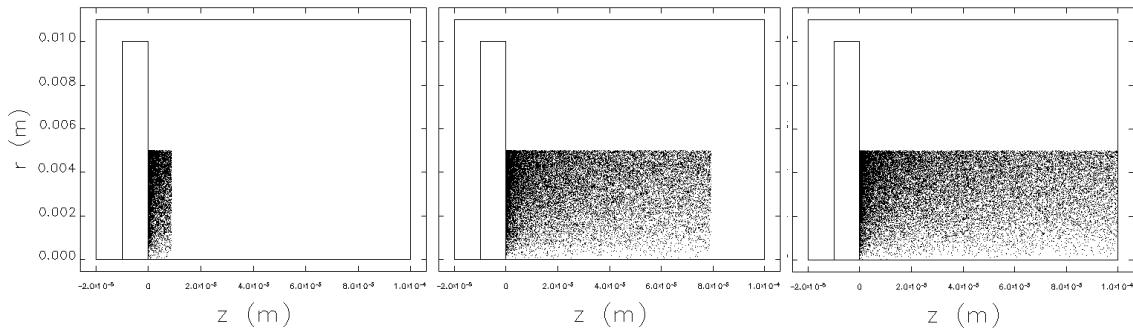


Figure 27. Right plate removed, electron beam acceleration.

With the space charge off, the results (Table 3) are consistent with the previous geometries.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
300	0	299.4907	6.256E-03	0	4.999995
	1	299.4482	1.399E-02	0	4.999995
	2	299.5006	1.877E-02	0	4.999995

Table 3. Right plate removed, space charge off.

With the space charge on, the current density delivered falls between the box geometry and full parallel plate geometry, and is the same (to 4 significant figures) as the left-plate- removed scenario. The results are listed in Table 4.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
300	0	218.2281	3.239E-03	3.611E-06	5.001601
	1	215.9259	1.132E-02	1.716E-05	5.011982
	2	215.3820	1.717E-02	1.648E-05	5.021624

Table 4. Right plate removed, space charge on.

C. PLATE THICKNESS VARIATIONS

Returning to the original parallel plate geometry, the plate thickness was varied and the results compared to the baseline from Chapter V. Four scenarios were evaluated, two with increased thickness and two with decreased thickness.

1. Increased Thickness

The plate thickness was increased from 0.01 mm to 0.015 mm. The cavity was left at its original width, giving a 0.005 mm margin between the plates and boundary. This simulation failed to produce current.

The simulation cavity was then widened by 0.005 mm on each side to give a 0.01 mm margin between the plates and boundary. The grid resolution remained at 0.005 mm, providing a grid line on each boundary of the plate and one internal.

With the space charge off, the plate width had no effect. The electric field was consistent with the plots shown in the parallel plate section and the current density delivered was identical to the original geometry. The results are listed in Table 5.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
300	0	299.4907	6.256E-03	0	4.999995
	1	299.4482	1.399E-02	0	4.999995
	2	299.5006	1.877E-02	0	4.999995

Table 5. Plate thickness 0.015 mm, space charge off.

With the space charge on, increasing the plate thickness had minimal effect on the current density. The results from the original geometry are shown in Table 6. The results from the thicker plates are shown in Table 7.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
300	0	187.3599	3.112E-03	4.479E-06	5.002438
	1	184.9992	1.083E-02	1.831E-05	5.015919
	2	185.8662	1.641E-02	1.830E-05	5.027015

Table 6. Plate thickness 0.01 mm (original geometry), space charge on.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
300	0	187.1261	3.115E-03	5.137E-06	5.002460
	1	185.0853	1.083E-02	1.839E-05	5.015656
	2	185.8173	1.642E-02	1.829E-05	5.026623

Table 7. Plate thickness 0.015 mm, space charge on.

Taking it a step further, the plates were widened to 0.02 mm. This time the plates had two internal grid lines. The plate width had negligible impact on the current density, as shown in Table 8.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
300	0	187.0780	3.115E-03	5.737E-06	5.002595
	1	185.3433	1.083E-02	1.885E-05	5.016203
	2	185.9997	1.641E-02	1.857E-05	5.027237

Table 8. Plate thickness 0.02 mm, space charge on.

2. Decreased Thickness

The plate thickness was decreased from 0.01 mm to 0.005 mm. The grid resolution was left at 0.005 mm, providing a grid line on each border of the plate but none internal. With the space charge off, narrowing the plate had no effect, as shown by the data in Table 9.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
300	0	299.4907	6.256E-03	0	4.999995
	1	299.4482	1.399E-02	0	4.999995
	2	299.5006	1.877E-02	0	4.999995

Table 9. Plate thickness 0.005 mm, space charge off.

With the space charge on, decreasing the plate thickness resulted in a decrease in current density of approximately 12%, as shown in Table 10.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
300	0	164.6370	3.039E-03	6.37E-06	5.003461
	1	164.6696	1.051E-02	1.96E-05	5.018811
	2	165.1050	1.592E-02	2.01E-05	5.030431

Table 10. Plate thickness 0.005 mm, space charge on.

The mesh was adjusted to provide a grid resolution of 0.0025 mm longitudinally (keeping 0.005 mm radially). This provided one internal grid line for each plate. This resulted in a slight decrease in current (Table 11), which can be attributed to the change in grid resolution.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
300	0	159.4560	3.363E-03	4.37E-06	5.003652
	1	159.4374	1.071E-03	1.51E-05	5.013048
	2	159.8853	1.602E-02	1.50E-05	5.022151

Table 11. Plate thickness 0.005 mm, space charge on, 0.0025 mm longitudinal grid resolution.

The plates were narrowed to 0.001 mm, resulting in another decrease in the current density. This data is shown in Table 12.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
300	0	144.9218	3.563E-03	2.943E-06	5.001991
	1	145.2404	1.066E-02	1.210E-05	5.011490
	2	145.6401	1.581E-02	1.230E-05	5.019882

Table 12. Plate thickness 0.001 mm, space charge on, 0.001 mm longitudinal grid resolution.

There seems to be a critical point in plate width above which there is negligible impact on the current density. Tables 9-12 are summarized in Figure 28.

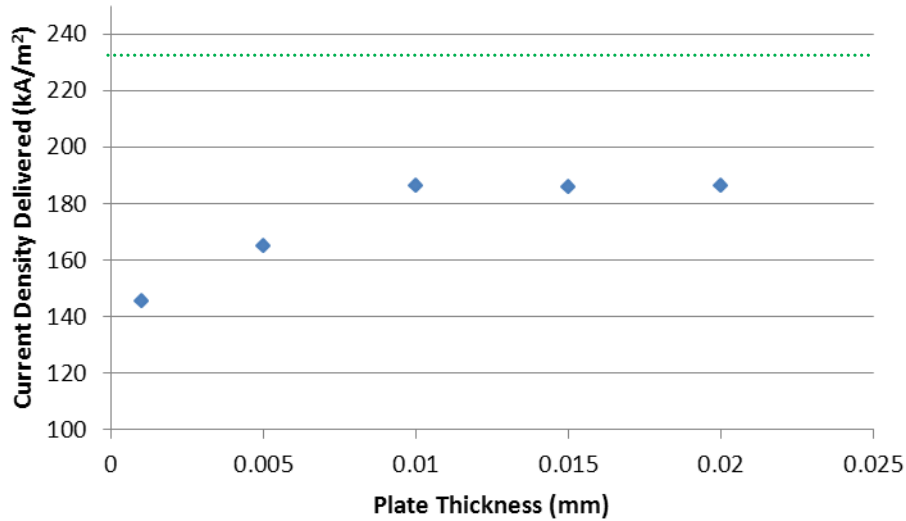


Figure 28. Variation of current density with plate thickness.

3. Cavity Size

The margin between the plate and cavity border in the original parallel plate geometry is 0.01 mm. When the plate thickness was increased, the cavity was widened by the same amount to keep the margin between plate and cavity border at 0.01 mm. When the plate was widened without adjusting the cavity width, no current density was delivered. This led to an investigation of the impact of cavity size. The plate thickness was set at the original value (0.01 mm) and the cavity was widened to provide a margin of 0.02 mm between each plate and the cavity boundary. There was little impact on the current provided, as shown in Table 13.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
300	0	187.4077	3.112E-03	4.577E-06	5.002438
	1	184.9943	1.083E-02	1.825E-05	5.015945
	2	185.8647	1.641E-02	1.840E-05	5.027093

Table 13. Wide cavity (0.02 mm margin), original plate thickness, space charge on.

D. PLATE RADIUS VARIATIONS

This section investigates the impact of plate radius on the saturation current density. The beam radius remained at 5 mm while the plate radius was varied.

1. Decreased Plate Radius

All dimensions were held constant while the plate radius was decreased from 10 mm to 7.5 mm. The beam radius remained at 5 mm. The reduction in plate radius caused a reduction in current density. The results from the original geometry are shown in Table 14. Table 15 lists the results from the decreased plate radius.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
300	0	187.3599	3.112E-03	4.479E-06	5.002438
	1	184.9992	1.083E-02	1.831E-05	5.015919
	2	185.8662	1.641E-02	1.830E-05	5.027015

Table 14. Original radius (10 mm), space charge on.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
300	0	147.7252	2.970E-03	4.777E-06	5.008343
	1	148.2254	1.024E-02	1.909E-05	5.026992
	2	148.1635	1.549E-02	1.993E-05	5.040493

Table 15. Decreased radius (7.5 mm), space charge on.

The radius/height of the cavity was also varied. As with the thickness variation, this showed negligible impact. The results, listed in Table 16, are similar to the results listed in Table 15.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
300	0	148.7674	2.985E-03	3.294E-06	5.006763
	1	146.9948	1.028E-02	1.779E-05	5.023468
	2	147.4970	1.551E-02	1.919E-05	5.033166

Table 16. Decreased radius (7.5 mm), decreased cavity height (8.5 mm).

2. Increased Radius

The radius of the plates was increased to 20 mm; the results are listed in Table 17. The current density increased, coming closer to the value obtained in the box geometry.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
300	0	241.3970	3.310E-03	3.847E-06	5.000949
	1	239.3932	1.168E-02	1.627E-05	5.009932
	2	238.6671	1.774E-02	1.461E-05	5.018646

Table 17. Increased plate radius (20 mm), space charge on.

Taking it a step further, the plate height was increased to 40 mm and then 60 mm; the results are shown in Table 18 and Table 19. As the plate height approaches some critical value, the parallel plate geometry comes very close to the box geometry.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
300	0	259.3666	3.395E-03	4.004E-06	5.000672
	1	256.5170	1.195E-02	1.579E-05	5.007867
	2	255.7882	1.815E-02	1.331E-05	5.014946

Table 18. Increased plate radius (40 mm), space charge on.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
300	0	263.0620	3.421E-03	3.992E-06	5.000683
	1	259.3737	1.200E-02	1.514E-05	5.007839
	2	258.4446	1.822E-02	1.267E-05	5.015056

Table 19. Increased plate radius (60 mm), space charge on.

The results from Tables 14-19 are summarized in Figure 29.

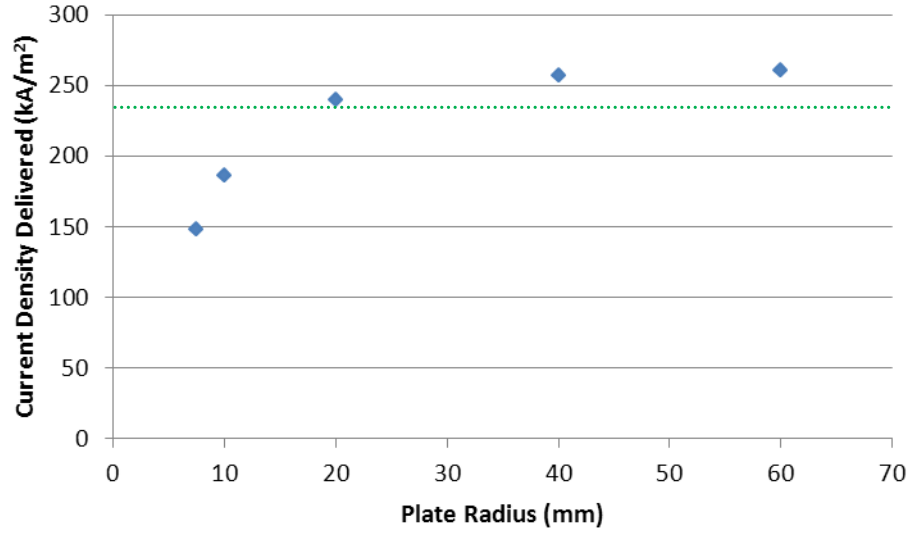


Figure 29. Variation of current density with plate radius.

THIS PAGE INTENTIONALLY LEFT BLANK

VII. SIMULATION TIME VARIATIONS

A. OVERVIEW

The previous simulations used a simulation time of 0.5 ns. This section investigates the impact of simulation time on the current density delivered.

B. BOX

If the simulation time is extended, the simple box geometry returns a value consistent with the original run time, whether the space charge is on or off. The results are summarized in Table 20 (space charge off) and Table 21 (space charge on).

Time (ns)	J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)
0.5	100	0	99.83024
		1	99.81605
		2	99.83352
1	100	0	99.83331
		1	99.84319
		2	99.84277
2	100	0	99.92712
		1	99.92187
		2	99.92384

Time (ns)	J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)
0.5	300	0	299.4907
		1	299.4482
		2	299.5006
1	300	0	299.4999
		1	299.5296
		2	299.5283
2	300	0	299.7814
		1	299.7656
		2	299.7715

Table 20. Varying simulation time, box, space charge off.

Time (ns)	J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)
0.5	100	0	99.51599
		1	99.31531
		2	99.28589
1	100	0	99.70458
		1	99.60492
		2	99.59658
2	100	0	99.86450
		1	99.80866
		2	99.80341

Time (ns)	J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)
0.5	300	0	270.2160
		1	267.0417
		2	266.0374
1	300	0	272.0603
		1	270.7792
		2	270.9607
2	300	0	268.5381
		1	268.4891
		2	268.7780

Table 21. Varying simulation time, box, space charge on.

C. PARALLEL PLATES

The parallel plate geometry is affected by the simulation time when the space charge is on. As the simulation time increases, the current density decreases. This effect is more noticeable when the current density is above the theoretical saturation current density. Table 22 shows the results when the space charge is off; Table 23 shows the results when the space charge is on.

Time (ns)	J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)
0.5	100	0	99.83024
		1	99.81605
		2	99.83352
1	100	0	99.83331
		1	99.84319
		2	99.84277
2	100	0	99.92712
		1	99.92187
		2	99.92384

Time (ns)	J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)
0.5	300	0	299.4907
		1	299.4482
		2	299.5006
1	300	0	299.4999
		1	299.5296
		2	299.5283
2	300	0	299.7814
		1	299.7656
		2	299.7715

Table 22. Varying simulation time, parallel plate, space charge off.

Time (ns)	J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)
0.5	100	0	99.08017
		1	98.39032
		2	98.02934
1	100	0	98.78264
		1	97.91741
		2	97.59198
2	100	0	80.78802
		1	80.70150
		2	80.70641

Time (ns)	J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)
0.5	300	0	187.3599
		1	184.9992
		2	185.8662
1	300	0	136.0386
		1	136.2075
		2	136.4122
2	300	0	85.26074
		1	85.48097
		2	85.66611

Table 23. Varying simulation time, parallel plate, space charge on.

VIII. BOX VARIATIONS

A. OVERVIEW

At this point, the parallel plate geometry and its variations are set aside. The highly variable response of this geometry to increased run times, plate radius, and plate thickness preclude setting a reasonable baseline for further study. The following simulations examine the impact of varying aspect ratio for the box geometry only. In the first set of simulations, the aspect ratio is varied by changing the interelectrode spacing. Next, the aspect ratio is held constant while the geometry is scaled up. In all cases, the voltage was adjusted to keep the electric field magnitude constant at 1 MV/m.

B. ASPECT RATIO VARIATIONS

Changing the aspect ratio by varying the interelectrode spacing also changes the theoretical saturation current density. Conversely, changing the aspect ratio by varying the cathode radius does not change the theoretical saturation current density. The theoretical saturation current changes, but since the current density is determined by dividing by the area, beam diameter is removed from the calculation.

1. AR=200

The cathode remained at 5 mm while the interelectrode spacing was decreased to 0.05 mm.

$$AR = \frac{2(5 \text{ mm})}{0.05 \text{ mm}} = 200 \quad (32)$$

The accelerating voltage was adjusted to 50 V to maintain the electric field strength at 1 MV/m between the electrodes. The theoretical saturation current density is 330 kA/m².

$$P = \frac{\pi \epsilon_0}{9} \sqrt{\frac{2q}{m}} (AR)^2 = \frac{\pi(8.85 \times 10^{-12})}{9} \sqrt{\frac{2(1.602 \times 10^{-19})}{9.109 \times 10^{-31}}} (200)^2 = 0.07328 \quad (33)$$

$$I = PV^{3/2} = (0.07328)(50)^{3/2} = 25.91 \text{ A} \quad (34)$$

$$J = \frac{I}{A} = \frac{25.91 \text{ A}}{\pi(5 \text{ mm})^2} = 330 \frac{\text{kA}}{\text{m}^2} \quad (35)$$

Figure 30 shows that the saturation current density delivered by spiffe is approximately 380 kA/m².

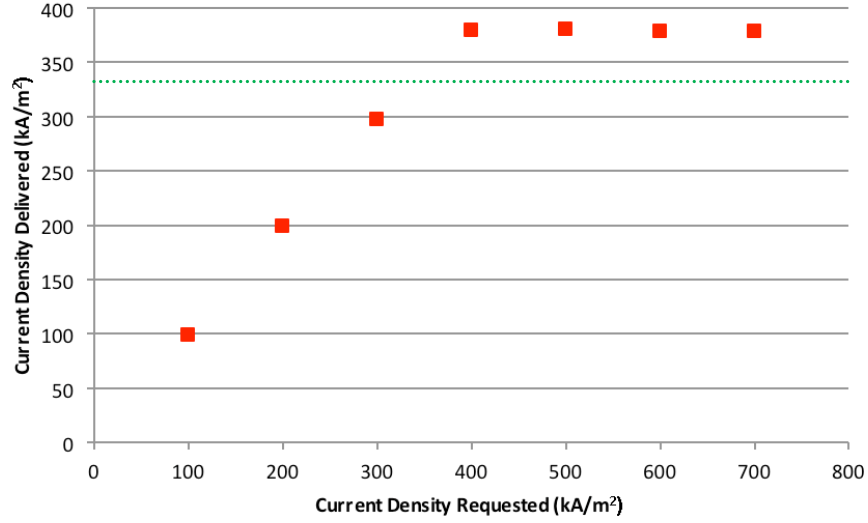


Figure 30. Box geometry, AR=200.

2. AR=50

The cathode remained at 5 mm while the interelectrode spacing was increased to 0.2 mm.

$$AR = \frac{2(5 \text{ mm})}{0.2 \text{ mm}} = 50 \quad (36)$$

The accelerating voltage was adjusted to 200 V to maintain the electric field strength at 1 MV/m between the electrodes. The theoretical saturation current density is 165 kA/m².

$$P = \frac{\pi\epsilon_0}{9} \sqrt{\frac{2q}{m}} (AR)^2 = \frac{\pi(8.85 \times 10^{-12})}{9} \sqrt{\frac{2(1.602 \times 10^{-19})}{9.109 \times 10^{-31}}} (50)^2 = 0.004580 \quad (37)$$

$$I = PV^{3/2} = (0.004580)(200)^{3/2} = 12.95 \text{ A} \quad (38)$$

$$J = \frac{I}{A} = \frac{12.95 \text{ A}}{\pi(5 \text{ mm})^2} = 165 \frac{\text{kA}}{\text{m}^2} \quad (39)$$

Figure 31 shows that the saturation current density delivered by spiffe is approximately 180 kA/m².

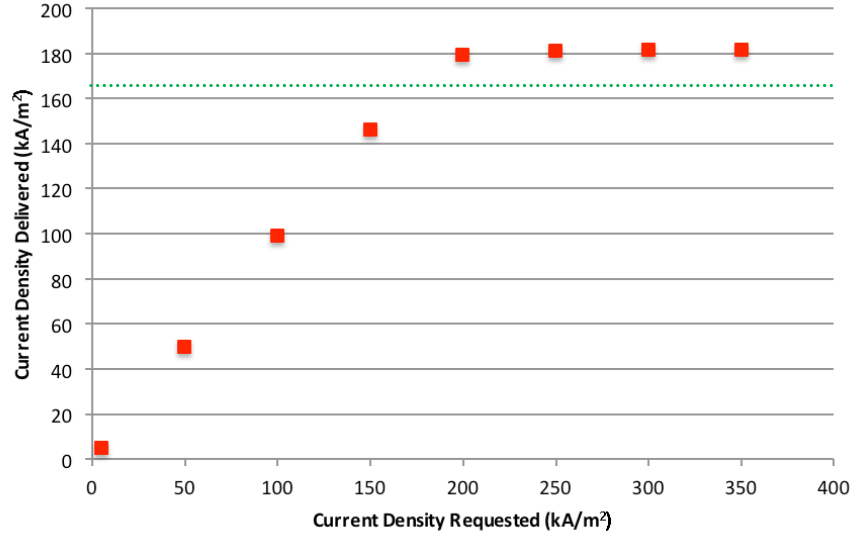


Figure 31. Box geometry, AR=50.

3. AR=10

The cathode remained at 5 mm while the interelectrode spacing was increased to 1 mm.

$$AR = \frac{2(5 \text{ mm})}{1 \text{ mm}} = 10 \quad (40)$$

The accelerating voltage was adjusted to 1000 V to maintain the electric field strength at 1 MV/m between the electrodes. The theoretical saturation current density is 73.8 kA/m².

$$P = \frac{\pi\epsilon_0}{9} \sqrt{\frac{2q}{m}} (AR)^2 = \frac{\pi(8.85 \times 10^{-12})}{9} \sqrt{\frac{2(1.602 \times 10^{-19})}{9.109 \times 10^{-31}}} (10)^2 = 0.0001832 \quad (41)$$

$$I = PV^{3/2} = (0.0001832)(1000)^{3/2} = 5.79 \text{ A} \quad (42)$$

$$J = \frac{I}{A} = \frac{5.79 \text{ A}}{\pi(5 \text{ mm})^2} = 73.8 \frac{\text{kA}}{\text{m}^2} \quad (43)$$

Figure 32 shows that the saturation current density delivered by spiffe is approximately 93 kA/m².

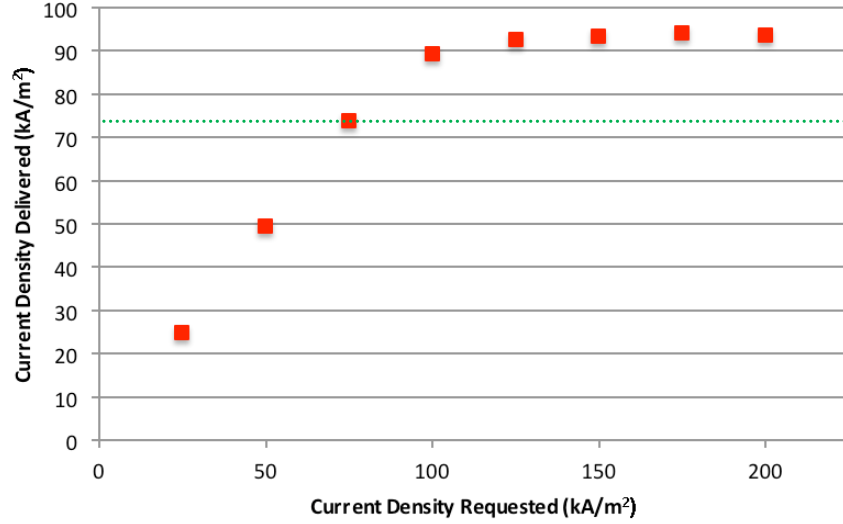


Figure 32. Box geometry, AR=10.

4. AR=5

The cathode remained at 5 mm while the interelectrode spacing was increased to 2 mm.

$$AR = \frac{2(5 \text{ mm})}{2 \text{ mm}} = 5 \quad (44)$$

The accelerating voltage was adjusted to 2000 V to maintain the electric field strength at 1 MV/m between the electrodes. The theoretical saturation current density is 52.2 kA/m².

$$P = \frac{\pi\epsilon_0}{9} \sqrt{\frac{2q}{m}} (AR)^2 = \frac{\pi(8.85 \times 10^{-12})}{9} \sqrt{\frac{2(1.602 \times 10^{-19})}{9.109 \times 10^{-31}}} (5)^2 = 4.58 \times 10^{-5} \quad (45)$$

$$I = PV^{3/2} = (4.58 \times 10^{-5})(2000)^{3/2} = 4.09 \text{ A} \quad (46)$$

$$J = \frac{I}{A} = \frac{4.09 \text{ A}}{\pi(5 \text{ mm})^2} = 52.2 \frac{\text{kA}}{\text{m}^2} \quad (47)$$

Figure 33 shows that the saturation current density delivered by spiffe is approximately 67 kA/m².

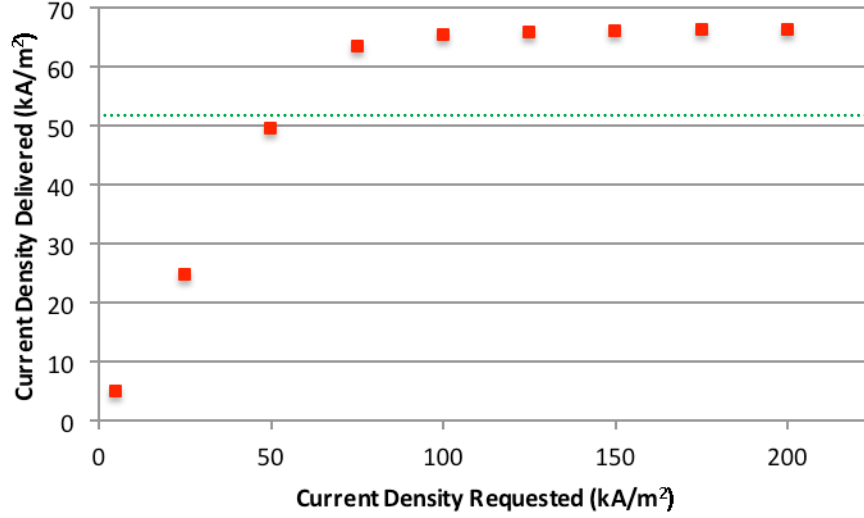


Figure 33. Box geometry, AR=5.

5. AR=1

The cathode remained at 5 mm while the interelectrode spacing was increased to 10 mm.

$$AR = \frac{2(5 \text{ mm})}{10 \text{ mm}} = 1 \quad (48)$$

The accelerating voltage was adjusted to 10,000 V to maintain the electric field strength at 1 MV/m between the electrodes. The theoretical saturation current density is 23.3 kA/m².

$$P = \frac{\pi \epsilon_0}{9} \sqrt{\frac{2q}{m}} (AR)^2 = \frac{\pi(8.85 \times 10^{-12})}{9} \sqrt{\frac{2(1.602 \times 10^{-19})}{9.109 \times 10^{-31}}} (1)^2 = 1.83 \times 10^{-6} \quad (49)$$

$$I = PV^{3/2} = (1.83 \times 10^{-6})(10000)^{3/2} = 1.83 \text{ A} \quad (50)$$

$$J = \frac{I}{A} = \frac{1.83 \text{ A}}{\pi(5 \text{ mm})^2} = 23.3 \frac{\text{kA}}{\text{m}^2} \quad (51)$$

Figure 34 shows that the saturation current density delivered by spiffe is approximately 40 kA/m².

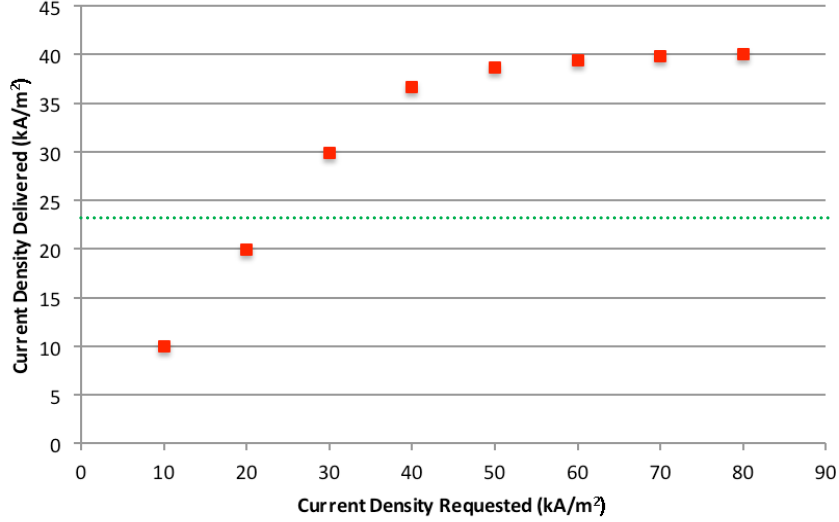


Figure 34. Box geometry, AR=1.

C. CONSTANT ASPECT RATIO, GEOMETRY SCALED-UP

In the following simulations, the aspect ratio is maintained at 100 by varying both the interelectrode spacing and the cathode radius. The theoretical current density changes because the interelectrode spacing has changed.

1. Scale Up ×5

The cathode radius was increased to 25 mm and the interelectrode spacing was increased to 0.5 mm. The grid resolution was adjusted to 0.025 mm.

$$AR = \frac{2(25 \text{ mm})}{0.5 \text{ mm}} = 100 \quad (52)$$

The accelerating voltage was adjusted to 500 V to maintain the electric field strength at 1 MV/m between the electrodes. The theoretical saturation current density is 104 kA/m².

$$P = \frac{\pi \epsilon_0}{9} \sqrt{\frac{2q}{m}} (AR)^2 = \frac{\pi(8.85 \times 10^{-12})}{9} \sqrt{\frac{2(1.602 \times 10^{-19})}{9.109 \times 10^{-31}}} (100)^2 = 1.83 \times 10^{-2} \quad (53)$$

$$I = PV^{3/2} = (1.83 \times 10^{-2})(500)^{3/2} = 205 \text{ A} \quad (54)$$

$$J = \frac{I}{A} = \frac{205 \text{ A}}{\pi(25 \text{ mm})^2} = 104 \frac{\text{kA}}{\text{m}^2} \quad (55)$$

Figure 35 shows that the saturation current density delivered by spiffe is approximately 118 kA/m².

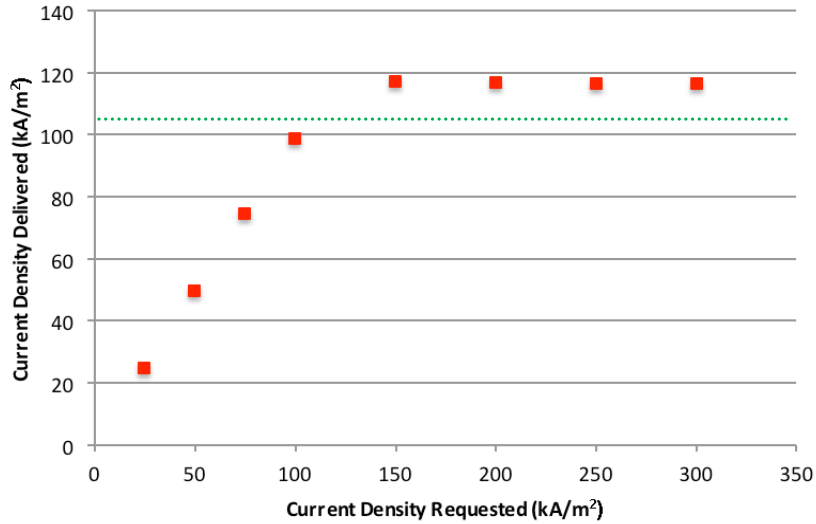


Figure 35. Box geometry, magnification $\times 5$

2. Scale Up $\times 10$

The cathode radius was increased to 50 mm and the interelectrode spacing was increased to 1 mm. The grid resolution was adjusted to 0.05 mm.

$$AR = \frac{2(50 \text{ mm})}{1 \text{ mm}} = 100 \quad (56)$$

The accelerating voltage was adjusted to 1000 V to maintain the electric field strength at 1 MV/m between the electrodes. The theoretical saturation current density is 73.7 kA/m².

$$P = \frac{\pi \epsilon_0}{9} \sqrt{\frac{2q}{m}} (AR)^2 = \frac{\pi(8.85 \times 10^{-12})}{9} \sqrt{\frac{2(1.602 \times 10^{-19})}{9.109 \times 10^{-31}}} (100)^2 = 1.83 \times 10^{-2} \quad (57)$$

$$I = PV^{3/2} = (1.83 \times 10^{-2})(1000)^{3/2} = 579 \text{ A} \quad (58)$$

$$J = \frac{I}{A} = \frac{579 \text{ A}}{\pi(50 \text{ mm})^2} = 73.7 \frac{\text{kA}}{\text{m}^2} \quad (59)$$

Figure 36 shows that the saturation current density delivered by spiffe is approximately 78 kA/m².

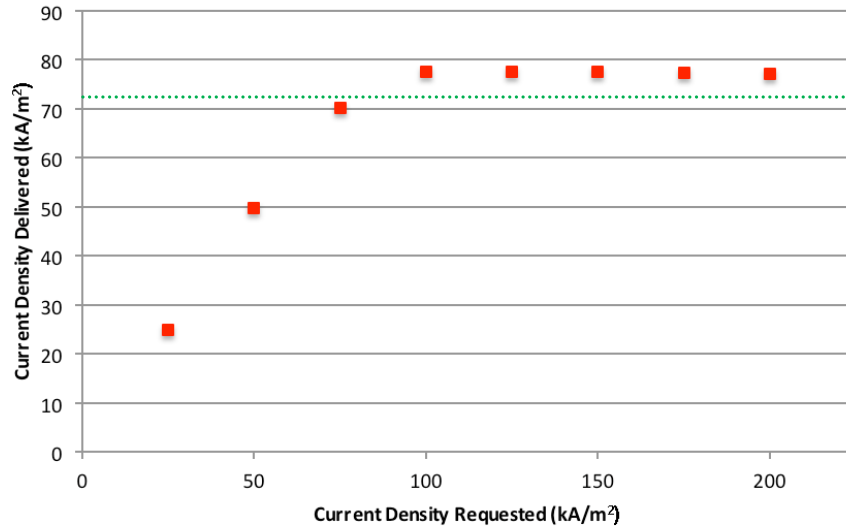


Figure 36. Box geometry, magnification $\times 10$

3. Scale Up $\times 50$

The cathode radius was increased to 250 mm and the interelectrode spacing was increased to 5 mm. The grid resolution was adjusted to 0.25 mm.

$$AR = \frac{2(250 \text{ mm})}{5 \text{ mm}} = 100 \quad (60)$$

The accelerating voltage was adjusted to 5000 V to maintain the electric field strength at 1 MV/m between the electrodes. The theoretical saturation current density is 33.0 kA/m².

$$P = \frac{\pi \epsilon_0}{9} \sqrt{\frac{2q}{m}} (AR)^2 = \frac{\pi(8.85 \times 10^{-12})}{9} \sqrt{\frac{2(1.602 \times 10^{-19})}{9.109 \times 10^{-31}}} (100)^2 = 1.83 \times 10^{-2} \quad (61)$$

$$I = PV^{3/2} = (1.83 \times 10^{-2})(5000)^{3/2} = 6477 \text{ A} \quad (62)$$

$$J = \frac{I}{A} = \frac{6477 \text{ A}}{\pi(250 \text{ mm})^2} = 33.0 \frac{\text{kA}}{\text{m}^2} \quad (63)$$

Figure 37 shows that the saturation current density delivered by spiffe is approximately 25 kA/m².

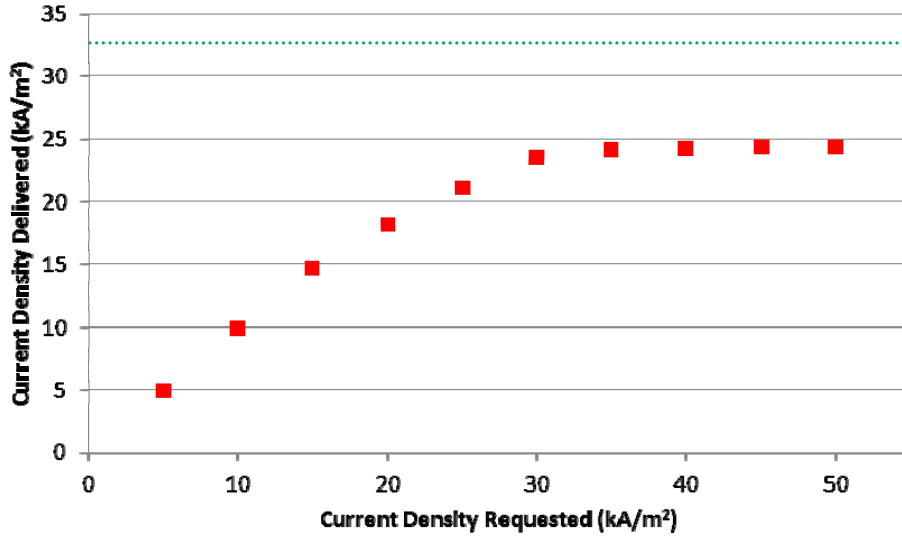


Figure 37. Box geometry, magnification $\times 50$

4. Scale Up $\times 100$

The cathode radius was increased to 500 mm and the interelectrode spacing was increased to 10 mm. The grid resolution was adjusted to 0.25 mm.

$$AR = \frac{2(500 \text{ mm})}{10 \text{ mm}} = 100 \quad (64)$$

The accelerating voltage was adjusted to 10,000 V to maintain the electric field strength at 1 MV/m between the electrodes. The theoretical saturation current density is 23.3 kA/m².

$$P = \frac{\pi \epsilon_0}{9} \sqrt{\frac{2q}{m}} (AR)^2 = \frac{\pi(8.85 \times 10^{-12})}{9} \sqrt{\frac{2(1.602 \times 10^{-19})}{9.109 \times 10^{-31}}} (100)^2 = 1.83 \times 10^{-2} \quad (65)$$

$$I = PV^{3/2} = (1.83 \times 10^{-2})(10000)^{3/2} = 18320 \text{ A} \quad (66)$$

$$J = \frac{I}{A} = \frac{18320 \text{ A}}{\pi(500 \text{ mm})^2} = 23.3 \frac{\text{kA}}{\text{m}^2} \quad (67)$$

Figure 38 shows that the saturation current density delivered by spiffe is approximately 14.5 kA/m².

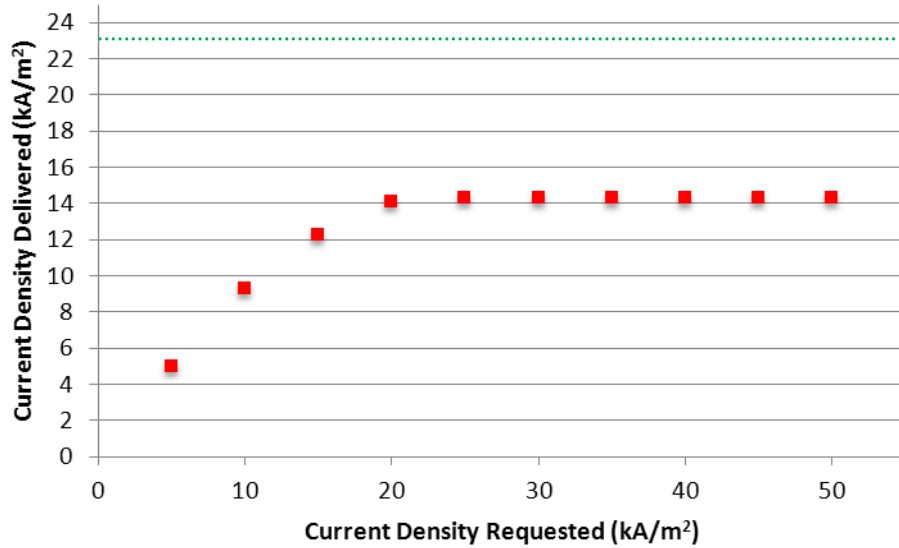


Figure 38. Box geometry, magnification $\times 100$.

IX. ION SIMULATION

A. PROGRAM SET-UP

`Spiffe` provides one option for altering the program to simulate ions. This is accomplished through use of the “stiffness” command. The stiffness is particle mass, given in terms of electron mass. For Argon, the stiffness is 72819, meaning that Argon is 72819 times as massive as an electron.

$$39.948 \text{ amu} \left(\frac{1.6605 \times 10^{-27} \text{ kg}}{\text{amu}} \right) \left(\frac{e^-}{9.1094 \times 10^{-31} \text{ kg}} \right) = 72819 e^- \quad (68)$$

There is no provision for changing the charge of the particle, so the problem must be set up in reverse. It is also helpful to increase the accelerating voltage. Since the particle is now significantly more massive, the acceleration will decrease for the same accelerating voltage (force) applied. This would require a longer simulation time for the particles to traverse the interelectrode spacing. In order to keep the simulation time at the same value used in the baseline simulations (0.5 ns), the potential was increased by the same order of magnitude as the stiffness. The potential of the original box geometry was increased from 100 V to 7 MV. If the potential had not been increased, the ion simulation would have to run 10^4 times longer to release the same number of macroparticles as the electron simulation. This begins to approach the maximum simulation time, limited by 32-bit signed integer memory.

B. RESULTS

With the accelerating voltage at 7 MV and AR=100, the theoretical saturation current density is 16 GA/m².

$$P = \frac{\pi \epsilon_0}{9} \sqrt{\frac{2q}{m}} (AR)^2 = \frac{\pi(8.85 \times 10^{-12})}{9} \sqrt{\frac{2(1.602 \times 10^{-19})}{6.633 \times 10^{-26}}} (100)^2 = 6.79 \times 10^{-5} \quad (69)$$

$$I = PV^{3/2} = (6.79 \times 10^{-5})(7 \times 10^6)^{3/2} = 1.26 \text{ MA} \quad (70)$$

$$J = \frac{I}{A} = \frac{1.26 \text{ MA}}{\pi(5 \text{ mm})^2} = 16 \frac{\text{GA}}{\text{m}^2} \quad (71)$$

Figure 39 shows that the saturation current density delivered by spiffe is approximately 18.5 GA/m^2 .

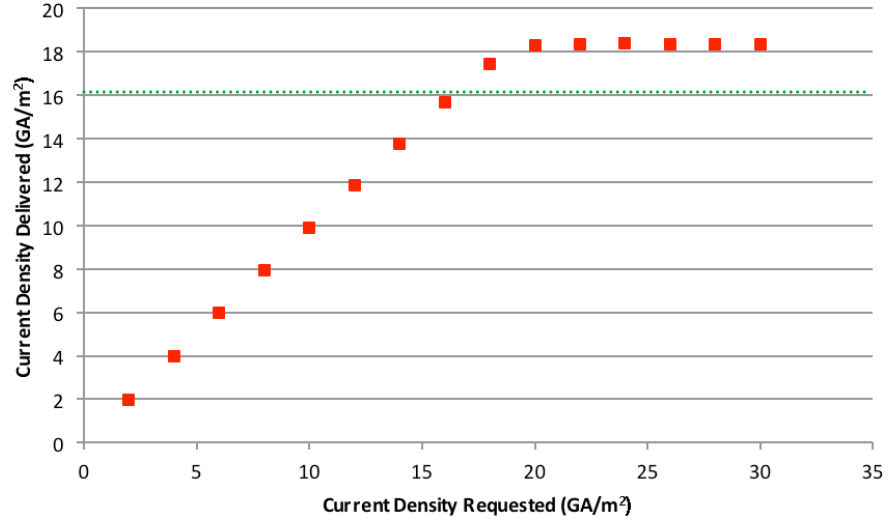


Figure 39. Ion simulation, box geometry, AR=100.

X. ANALYSIS

A. ONE-DIMENSIONAL APPROXIMATION

`Spiffe` accurately models the theoretical saturation current density in a one-dimensional approximation. The current density delivered by the simulation is approximately 15% higher than the theoretical value.

The slightly higher value of current density is expected due to beam spreading. The Child-Langmuir law applies to infinite parallel plates. In a beam of finite width, the space charge interaction between particles will cause beam spreading. The high aspect ratio of the box approximates a one-dimensional situation, but because the beam is finite, it will still experience spreading.

The electric field from a point charge is a function of $1/r^2$. In a two-dimensional scenario with a beam of finite width, interactions between the ions cause the beam to spread radially. As the beam spreads (i.e., r increases), the effect of the ions' electric field at the source is diminished. Since the electric field from the ions counteracts the external electric field, a decrease in the ion electric field allows an increase in the external electric field. The increased external electric field allows the source to emit a current density that is higher than the theoretical limit given by the Child-Langmuir law.

Beam spreading in the parallel plates is greater, which implies a higher saturation current density, but this was not the result. The average longitudinal momentum in the parallel plate geometry is lower than that seen in the box geometry. This indicates that although the interelectrode spacing and applied potentials are the same, the parallel plate geometry is not accelerating particles as much as the box geometry.

In both the box and parallel plate geometries, with the space charge off, `spiffe` returns the expected current density and electric field distribution. With the space charge on, the geometries vary significantly. The box geometry returns a higher saturation current density than expected, which is explained by beam spreading. In contrast, the parallel plate geometry returns a much lower current density than expected. The electric

field plots provide some insight into this difference. Before particles are released, the equipotential lines of the box geometry are uniformly spaced. The equipotential lines of the parallel plate geometry are concentrated at the cathode and anode. The magnitude of the electric field at the centerline ($r=0$) of the box ranges from 4×10^5 V/m to 1.4×10^6 V/m. The range in the parallel plate geometry begins at 4×10^5 V/m but only increases to 1×10^6 V/m. This is perhaps why the particles are not being accelerated as in the box geometry. The cause for these differences is likely due to the difference in boundary conditions for the cavity border and the physical surface inside the cavity. Both boundary conditions require $E_z=0$, but the program interprets them differently when the space charge calculation is involved.

Removal of one plate from the parallel plate geometry confirms that it is the boundary conditions causing the differences between the box and parallel plate geometries. The variability of current with plate thickness and radius is another indicator that the boundary conditions are the cause for variations between the geometries.

As the plate radius increases, it approaches the values provided by the box geometry. The excessive radius needed to reach this situation is unnecessary when the simpler box geometry can accomplish the same result. Variations between the box and parallel plate geometries illustrate that the cathode should coincide with a cavity border.

B. BOX VARIATIONS

In the one-dimensional situation, the current density was approximately 15% higher than the theoretical limit. As the aspect ratio is decreased, the current density deviates further from the theoretical limit (Table 24, Figure 40). This is not necessarily an error in the modeling, but a result of the fact that beam spreading becomes significant at lower aspect ratios. The practical aspect ratio limit for a beam is on the order of unity, as noted in [2].

	AR					
	200	100	50	10	5	1
$J_{\text{theoretical}}$ (kA/m ²)	329.89	233.26	164.94	73.76	52.16	23.33
$J_{\text{delivered}}$ (kA/m ²)	381.25	268.64	181.54	93.54	66.19	39.50
$J_{\text{del}}/J_{\text{th}}$	1.16	1.15	1.10	1.27	1.27	1.69

Table 24. Theoretical vs. delivered current density, varying AR.

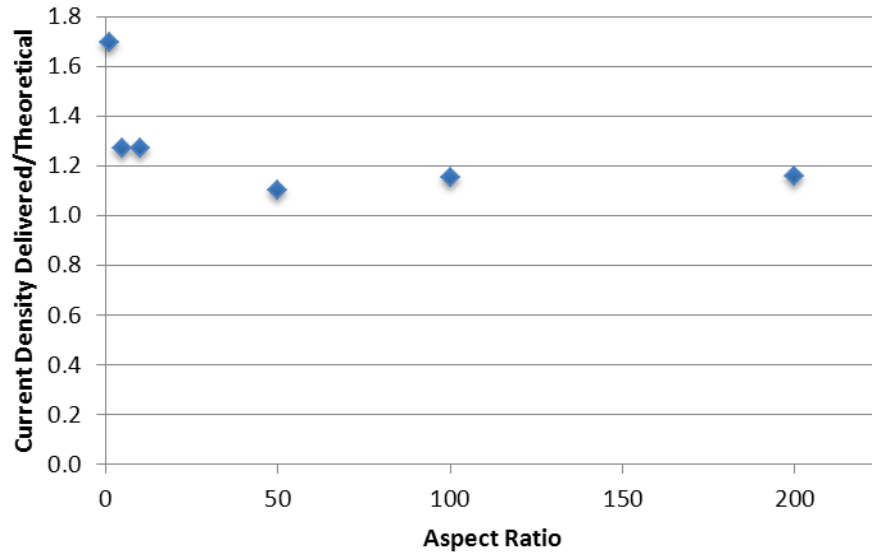


Figure 40. Variation in current density delivered/theoretical for varying aspect ratios.

`spiffe` is best used for small-scale geometries. Table 25 compares the current density for the magnified geometries. With the aspect ratio at 100 in all cases, `spiffe` should have returned a current density 15% greater than the theoretical limit. As the scaling of the geometry increased, the current density decreased (Table 25, Figure 41).

AR=100	Magnification				
	1	5	10	50	100
$J_{\text{theoretical}}$ (kA/m ²)	233.3	104.3	73.76	32.99	23.33
$J_{\text{delivered}}$ (kA/m ²)	268.6	117.1	77.65	24.37	14.37
$J_{\text{del}}/J_{\text{th}}$	1.15	1.12	1.05	0.74	0.62

Table 25. Theoretical vs. delivered current density, AR=100, geometry magnified.

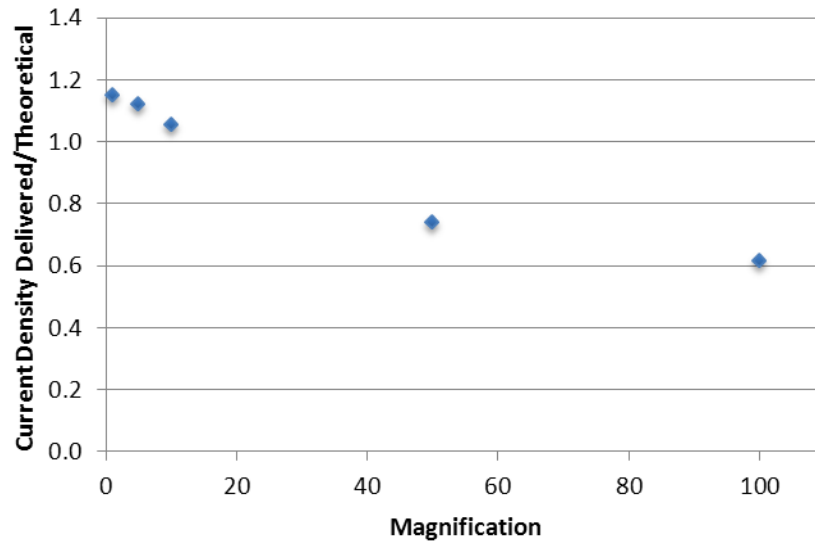


Figure 41. Variation in current density delivered/theoretical for varying magnifications, AR=100.

C. ION SIMULATION

With AR=100 and the simulation adjusted to simulate ions, the saturation current density is approximately 15% higher than the theoretical value. This is the same ratio that was seen for the electron simulation. The voltage was increased significantly to reduce simulation time. Voltage breakdown is a concern, but not at this stage.

XI. CONCLUSION AND RECOMMENDATIONS

A. CONCLUSION

`Spiffe` differs from current steady-state ion numerical modeling schemes in that it solves Maxwell's equations and performs a Poisson correction. As the program was intended to simulate electrons, many quantities are normalized to the mass and charge of an electron.

The code is able to simulate ions by adjusting the mass of the particle, but it provides no provision for changing the charge. This limits the simulation particles to singly-charged ions. This is satisfactory for field-ionization, which produces only singly-charged ions. The sign of the charge remains negative, so the user must be prepared to set-up the problem in reverse.

`Spiffe` accurately models a theoretical one-dimensional situation. Finite beam spreading causes the program to output a current density approximately 15% higher than the theoretical limit. As the geometry deviates from one-dimensional, the error increases due to radial beam spreading. This is not necessarily an error in the modeling, but a result of the fact that beam spreading becomes significant at lower aspect ratios.

A CNT field-ionized ion thruster model should be capable of modeling the electrode spacing and applied potentials to both interact with the CNTs to create ions and also to accelerate those ions. `Spiffe` is not capable of doing this for two reasons. The first is that `spiffe` cannot solve the Fowler-Nordheim equation, making it unable to model ionization of the propellant gas. Second, the user would be required to program the geometry of every nanotube, and the grid would have to be small enough to resolve those nanotubes and fill the entire accelerating region. The time step required for stability would be small and the simulation time would be long, both contributing to impractical run times and CPU memory limitations.

`Spiffe` is a useful tool for the early modeling stages of a CNT field-ionized ion thruster. It can be used to predict the saturation current density for the setup described in

[8]. Additionally, it can be used to model the ideal electrode spacing for a single beamlet with a simplified planar ion source. Since it is strictly an axi-symmetric code, `spiffe` is not able to model a multi-aperture grid. As the CNT field-ionized ion thruster is further developed, a more robust model should be pursued.

B. FUTURE WORK

Future work includes using `spiffe` to predict the current producing capability of the setup described in [8].

`Spiffe` may also be used for determining the thrust and perveance for a single beamlet in order to conduct feasibility studies for CNT field-ionized thrusters and micro- and nano-scale satellites.

Finally, future work must investigate alternate modeling options for full-scale numerical modeling of CNT field-ionized thrusters for micro- and nano-scale satellites.

APPENDIX A. SUPPLEMENTAL USER'S GUIDE

A. USER'S GUIDE

Reference [14] is the user's guide published by Argonne National Laboratory (ANL)/Advanced Photon Source (APS). It is available for download from the ANL website. This Appendix is intended to supplement the official user's guide to provide additional guidance for first-time users.

B. SDDS TOOLKIT

SDDS Toolkit is a set of commands that allows plotting and processing of the data obtained from `spiffe` runs. The toolkit and its user's guide are available for download from the ANL/APS website.

C. DOWNLOAD AND INSTALL

1. Download Address.

The ANL/APS software download page is located at:

http://www.aps.anl.gov/Accelerator_Systems_Division/Accelerator_Operations_Physics/software.shtml

2. Spiffe

Scroll down to the `spiffe` section and download the appropriate file for your operating system. The simplest method is to use the self-installing file (`Spiffe.exe`).

3. SDDS Toolkit

Scroll down to SDDS Binaries and download the appropriate file for your operating system. The simplest method is to use the self-installing file (`SDDS ToolKit.exe`).

4. Verify Installation

Once the `spiffe` installation is complete, open a command prompt window and type “`spiffe`” at the prompt. If installed correctly, the prompt will return the following lines:

```
Link date: Sep 6 2011 08:52:07 (or date of latest revision)
error: no input file listed
usage: spiffe inputfile
Program by M. Borland (This is version 4.1, August 2011.)
```

D. PROGRAM STRUCTURE

`Spiffe` files may be written in any text editing program. A `spiffe` run consists of at least two files – the geometry file and the main run file. The geometry file is an input to the run file. The run file may also output data to several files, which are specified by the user. The run file must have a “.spiffe” extension. The geometry file and all output files specified by the user may be given any extension. Common practice is to use something that indicates the type of data contained in the file. For example, “.geo” for the geometry file and “.snap” for the beam snapshot data. The recommended sequence of commands for the run file is given in [14].

E. GEOMETRY FILE

1. Overview

The geometry file can be given any name and any extension. The geometry file defines the geometry using a series of points. Each point may be assigned a potential. If a potential is not specified, the default is 0 V. Any geometry that can be defined with line segments and arcs can be defined with the geometry file. The most important thing to remember when defining a geometry file is that `spiffe` is an axisymmetric code. The geometry file uses x - y coordinates to define each point. When translated by the run file, x -coordinates correspond to the z -axis and y -coordinates correspond to radius.

The geometry file may be written using any unit of length, but meters is preferred for simplicity. If another unit is used, the `spiffe` run file provides a command line

(*lr_factor*) to convert units to meters.

An optional command that is not included in [14] is *ramp_potential*. This command allows you to ramp the assigned potential along the surface to prevent discontinuities. In the simple geometries used in this report, *ramp_potential* made little difference in the results. It may become important when defining boundaries inside the cavity.

2. Example – Box Geometry

The structure of the geometry file is described in Section 4.2 of [14]. This example illustrates how to create the box geometry used in this report.

First, open a new text editor file and name it *box.geo*. Type the following code into the text file:

```
&po x=0.0,      y=0,      potential=0,                      &end
&po x=0.0,      y=0.01, potential=0,    ramp_potential=1, &end
&po x=0.0001,   y=0.01, potential=100, ramp_potential=1, &end
&po x=0.0001,   y=0.0,   potential=100,                      &end
&po x=0.0,      y=0.0,   potential=0,                      &end
```

Each line represents a new point. Note that *&po* is an acceptable abbreviation for *&point*. Five points are defined because it is points that are defined, not line segments. If the first point was not restated, the fourth line segment would not be drawn. *Ramp_potential* is not used at $y=0$ since this is the centerline ($r=0$) of the axisymmetric beam.

F. RUN FILE

1. Overview

The run file calls on the geometry file, draws the geometry, and defines all aspects of the simulation (e.g., cathode radius, run time, integration step size, grid size, etc.)

The first step in creating a *spiffe* run file is to establish the geometry for the simulation. This is done with the geometry file and the *define_geometry* command, described in section 4.1 of [14] and discussed below.

2. Define the geometry

a. Cavity

The simulation takes place in a cavity. In the case of the box geometry, the simulation cavity is the box. In the case of the parallel plate geometry, the cavity is the region surrounding the plates. The simulation cavity is defined by specifying the minimum and maximum longitudinal coordinates (z_{\min} , z_{\max}) and the maximum radial coordinate (r_{\max}). The minimum radial coordinate is always $r=0$ since `spiffe` is an axisymmetric code.

b. Grid

The grid is defined by defining the cavity (z_{\min} , z_{\max} , r_{\max}) and specifying the number of grid lines (n_z , n_r). Grid lines are equally spaced between the minimum and maximum values; the shape is a simple quadrilateral. Radial and longitudinal grid resolution should be kept within a factor of two for best results.

A knowledge of the grid layout is essential for drawing geometries and assigning the proper number of grid lines. It is also helpful for interpretation of electric field plots and understanding how the space charge is calculated.

Field quantities are calculated using four interleaving grids. The grid and position-index correspondences are described in [12]. Figure 42 illustrates the relationship between the four interleaving grids. Black lines indicate the grid that is specified by the user with n_z and n_r , which is also the grid for calculating ρ and Φ . The longitudinal components of the electric field (E_z) and the current density (J_z) correspond to the red grid (horizontal lines overlap black). The radial components of the electric field (E_r) and current density (J_r) correspond to the blue grid (vertical lines overlap black). Finally, the grid for the magnetic field (B_ϕ) overlaps the red and blue lines.

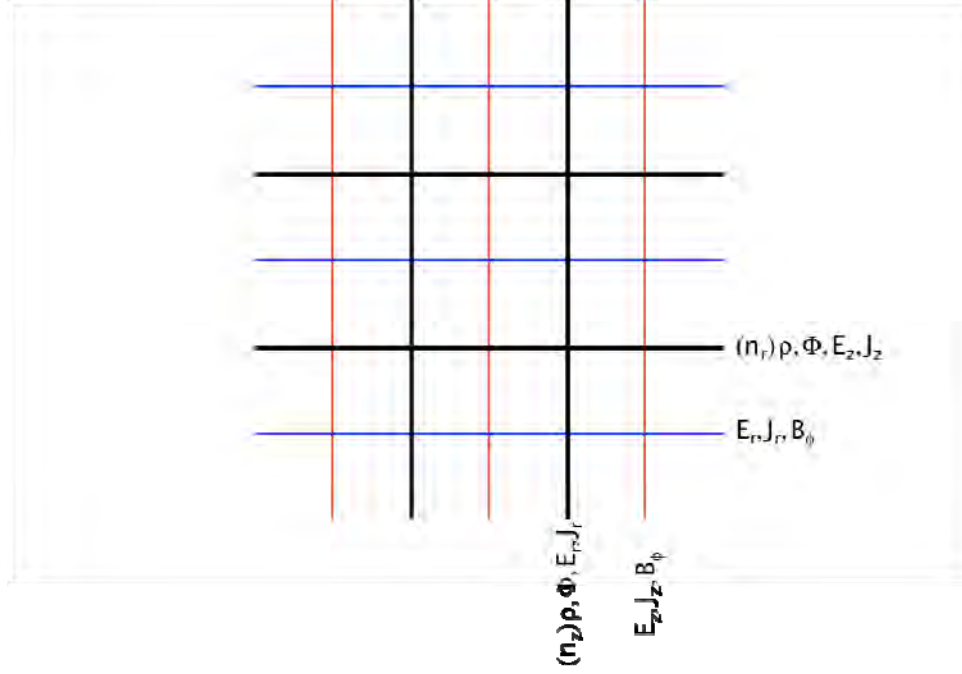


Figure 42. Interleaving grid.

In the box geometry, the only dimensions of concern are the length (0.1 mm) and radius (10 mm) of the simulation region. The longitudinal and radial grid spacing are defined in [12] as:

$$\Delta z = \frac{z_{\max} - z_{\min}}{n_z - 1} \quad (72)$$

$$\Delta r = \frac{r_{\max} - r_{\min}}{n_r - 1} \quad (73)$$

For a grid resolution of $\Delta z = \Delta r = 0.05 \text{ mm}$,

$$n_z = \frac{0.1 \text{ mm}}{0.05 \text{ mm}} + 1 = 3 \quad (74)$$

$$n_r = \frac{10 \text{ mm}}{0.05 \text{ mm}} + 1 = 201 \quad (75)$$

With this grid size, `spiffe` requires a time step less than 1.179327×10^{-13} sec for stable integration.

Table 26 compares `spiffe` output for various grid sizes and current density. The grid size has minimal impact when the requested current density is less than the saturation current density. The saturation current for the box geometry is approximately 233 kA/m^2 . With the largest grid resolution (fewest number of grid lines), the simulation returns a value well beyond the space charge limit. As the resolution improves (grid lines increase), `spiffe` provides a better approximation of the space charge limit.

Resolution	50 μm		25 μm		10 μm		5 μm	
z grid lines (nz)	3		5		11		21	
r grid lines (nr)	201		401		1001		2001	
Maximum time step	1.18E-13		5.90E-14		2.36E-14		1.18E-14	
Time step used	1E-13		5E-14		2E-14		1E-14	
Number of steps	5000		10000		25000		50000	
$J_{\text{req}}=100 \text{ kA/m}^2$	J_{del}	r_{max}	J_{del}	r_{max}	J_{del}	r_{max}	J_{del}	r_{max}
Screen 0	99.855	4.999998	99.288	4.999986	99.674	5.000044	99.517	5.000102
Screen 1	99.908	5.000202	99.086	5.000581	99.508	5.001042	99.312	5.001586
Screen 2	99.950	5.000606	99.080	5.001537	99.429	5.002419	99.283	5.003382
$J_{\text{req}}=300 \text{ kA/m}^2$	J_{del}	r_{max}	J_{del}	r_{max}	J_{del}	r_{max}	J_{del}	r_{max}
Screen 0	299.04	5.000016	296.32	5.000058	291.53	5.000261	270.14	5.000620
Screen 1	298.42	5.000761	293.55	5.002499	288.34	5.004768	266.93	5.007441
Screen 2	298.27	5.002157	293.47	5.005997	287.49	5.010420	265.92	5.014438
$J_{\text{req}}=500 \text{ kA/m}^2$	J_{del}	r_{max}	J_{del}	r_{max}	J_{del}	r_{max}	J_{del}	r_{max}
Screen 0	497.64	5.000045	405.78	5.000223	307.71	5.000858	271.10	5.001814
Screen 1	494.12	5.001426	402.27	5.007025	300.65	5.012518	268.08	5.013434
Screen 2	490.98	5.004336	398.27	5.015544	299.61	5.022965	266.73	5.023683

Table 26. Impact of grid size on simulation accuracy.

There is a tradeoff between simulation time and grid resolution. As the grid lines increase, the time step required for stability decreases, which increases the run

time of the simulation. In the simplest case (3×21), the simulation took less than 10 seconds. With the grid size used throughout this report, simulations averaged 20 to 40 minutes.

In the parallel plate geometry, the smallest dimension is the plate thickness at 0.01 mm (10 μm). The grid must be able to distinguish plate boundaries, so it must have a minimum resolution of 10 μm. The grid must also provide enough resolution to accurately model the particle and field interactions between the plates. The gap between the plates is 0.1 mm; a resolution of 10 μm would place ten grid lines between the plates.

The cavity surrounding the plates is 0.14 mm wide and 10 mm tall. Using a resolution of 10 μm (needed to distinguish plate thickness) and Equations (72) and (73), the number of longitudinal and radial grid lines is calculated by:

$$nz = \frac{0.14 \text{ mm}}{0.01 \text{ mm}} + 1 = 15 \quad (76)$$

$$nr = \frac{10 \text{ mm}}{0.01 \text{ mm}} + 1 = 1001 \quad (77)$$

When using this grid resolution, `spiffe` reports that the time step must be less than 2.540089×10^{-14} for stable integration.

c. Geometry Output Files

`Spiffe` writes two output files regarding the geometry. The first file is the geometry as it is specified by the user (`boundary_output`). The second file is the geometry as it is interpreted by `spiffe` (`discrete_boundary_output`). These files are helpful for verifying that the geometry was programmed or interpreted correctly.

It is essential to choose grid lines that coincide with boundaries. This can be verified by comparing the ideal boundary (user-defined) to the discrete boundary (`spiffe`-interpreted). For example, in the parallel plate model, the grid lines match up

with the plate boundaries (Figure 43). If the grid lines were chosen poorly, they would not line up with the plate boundaries and `spiffe` would not “see” the plate correctly. This is illustrated in Figure 44.

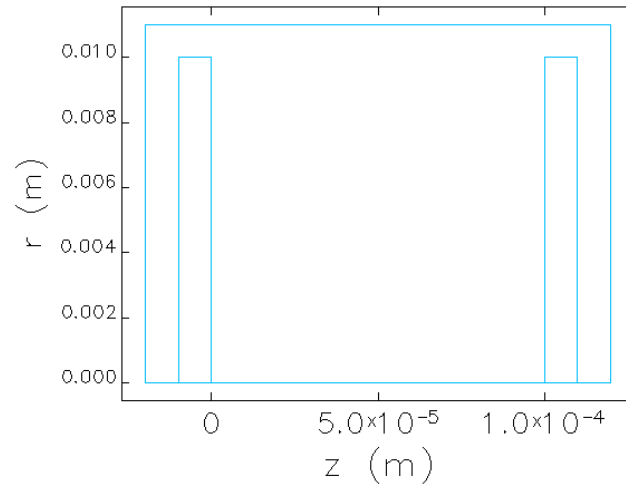


Figure 43. Geometry interpretation with correct number of grid lines.

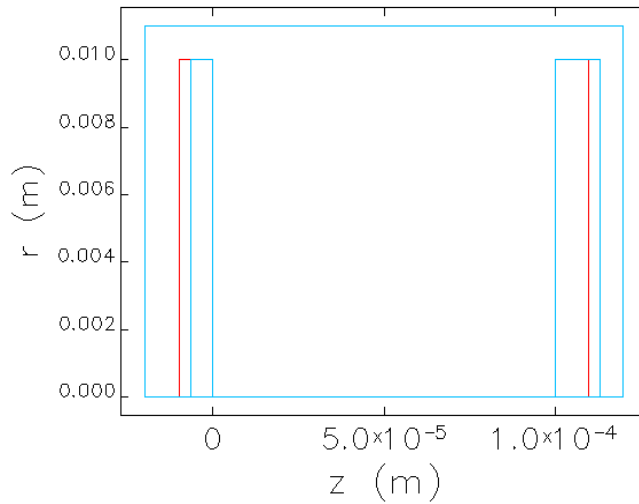


Figure 44. Geometry interpretation with incorrect number of grid lines.

SDDS Toolkit is needed to produce the plots in Figure 43 and Figure 44. If the ideal boundary file is named *duo.bnd* and the discrete boundary file is named *duo.dbnd*, the following command will produce the comparison plot.

```
sddsplot -col=z,r -graph=line,type=2 -split=page duo.bnd -
col=z,r -graph=line,type=3 -split=page duo.dbnd
```

Further details of the SDDS plot tools can be found in [15].

d. Boundary Conditions

Boundary conditions for the cavity are specified in the `define_geometry` section of the code. Upper, lower, left, and right boundary conditions apply to the boundaries of the cavity.

Boundary conditions for surfaces inside the cavity (e.g., plates) are assumed to be metal surfaces. In this case, $E_z=0$ for vertical surface and $E_r=0$ for horizontal surfaces. In other words, there are no electric field lines normal to the metal surfaces.

e. Radial and Longitudinal Interpolation

`Radial_interpolation` and `longitudinal_interpolation` should be set to “1” to indicate that field components should be interpolated when fields are applied to particles. This is necessary to prevent an abrupt change as particles move from one grid square to the next [14].

f. Example Code

The code used for the box geometry is included here. The code is written by simply opening a new text file and copying the text shown. The unit of measurement for everything specified in a `spiffe` run file is meters. The command `zr_factor` is a scaling factor that is used to convert the geometry file (*box.geo*) into meters. Since *box.geo* is already written in meters, the command `zr_factor` can be set to “1” or simply omitted from the code. Any items not included in the code are set to their default value. Additional code options and their default values are listed in Section 4.1 of [14].

```
&define_geometry
    nz=21,
    zmin=0,
```

```

        zmax=0.0001,
        nr=2000,
        rmax=0.01,
        boundary="box.geo",
        radial_interpolation=1,
        longitudinal_interpolation=1,
        boundary_output="%s.bnd",
        discrete_boundary_output="%s.dbnd",
        lower="Dirichlet",
        upper="Neumann",
        right="Neumann",
        left="Neumann",
&end

```

If the name of the spiffe run file is *box.spiffe*, then the notation *%s* will send the ideal boundary points to *box.bnd* and the discrete boundary points to *box.dbnd*. The parallel plate geometry was created with the following code.

```

!Define the cavity
&po x=-0.00002, y=0.0, potential=0, &end
&po x=-0.00002, y=0.011, potential=0, &end
&po x= 0.00012, y=0.011, potential=0, &end
&po x= 0.00012, y=0.0, potential=0, ramp_potential=1, &end
&po x= 0.00011, y=0.0, potential=100, ramp_potential=1, &end
&po x= 0.00010, y=0.0, potential=100, &end
&po x= 0.0, y=0.0, potential=0, &end
&po x=-0.00002, y=0.0, potential=0, &end

!Define the left plate
&po nt=3, x=0.0, y=0.0, potential=0, &end
&po x=0.0, y=0.01, potential=0, &end
&po x=-0.00001, y=0.01, potential=0, &end
&po x=-0.00001, y=0.0, potential=0, &end
&po x=0.0, y=0.0, potential=0, &end

!Define the right plate
&po nt=3, x=0.0001, y=0.0, potential=100, &end
&po x=0.0001, y=0.01, potential=100, &end
&po x=0.00011, y=0.01, potential=100, &end
&po x=0.00011, y=0.0, potential=100, &end
&po x=0.0001, y=0.0, potential=100, &end

```

3. Define the Cathode

This section of the code is where the user specifies the location of the electron

source, the current density requested, and the properties of the particles when they initially appear in the simulation (e.g., initial momentum).

a. Position

The physical cathode is defined in the geometry file, but there is no requirement to place the particles on the physical surface. Particles may be loaded from any location that can be described by radius and longitudinal position. Placing the cathode slightly off the physical surface theoretically ensures that no particles are artificially lost to the boundary. In reality, the cathode position (`z_position=0` versus `z_position=1e-8`) had minimal impact on the current delivered. As the size of the cathode increases, this may become more significant.

b. Current Density

The user can request any current density. If the value requested is greater than the saturation current, `spiffe` will not exceed the saturation current (within numerical error and considering finite radius beam spreading). Particles are generated randomly from the location specified.

c. Macroparticles

`Spiffe` is a particle-in-cell (PIC) code. PIC codes are characterized by the use of macroparticles, which represent several physical particles. The user may specify either `electrons_per_macroparticle` or `number_per_step`. `Electrons_per_macroparticle` tells `spiffe` how many electrons to assign to each macroparticle. For example, if the user specifies 500 electrons per macroparticle with a time step of 1×10^{-14} sec, a current density of 5000 A/m^2 , and a cathode radius of 5 mm, `spiffe` will calculate the appropriate number of macroparticles (*mp*) to release per time step.

$$5000 \frac{\text{A}}{\text{m}^2} \cdot \frac{\pi(5 \times 10^{-3} \text{m})^2}{\text{A} \cdot \text{s} / \text{C}} \cdot \frac{e^-}{1.602 \times 10^{-19} \text{C}} \cdot \frac{1 \times 10^{-14} \text{s}}{\text{step}} \cdot \frac{\text{mp}}{500 e^-} = 49 \frac{\text{mp}}{\text{step}} \quad (78)$$

If instead the user specifies the number of macroparticles to release per time step, `spiffe` will calculate the number of electrons that are represented by each macroparticle.

$$5000 \frac{A}{m^2} \cdot \frac{\pi(5 \times 10^{-3} m)^2}{A \cdot s / C} \cdot \frac{e^-}{1.602 \times 10^{-19} C} \cdot \frac{1 \times 10^{-14} s}{step} \cdot \frac{step}{8 mp} = 2064 \frac{e^-}{mp} \quad (79)$$

The choice is entirely up to the user. Fewer electrons per macroparticle will improve accuracy slightly but drastically increase computation time and memory.

d. Start/Stop Time

These times apply only to the release of particles. The simulation time is specified in a different section and does not need to match cathode start and stop times. Generally, it is recommended to start the cathode shortly after the simulation starts. This allows the initial external electric field to be established before releasing particles into the cavity.

e. Initial Momentum

The user selects the initial momentum, both longitudinal (`init_pz`) and radial (`init_pr`), to give the particles once they are randomly generated. This option can have a significant impact on the results and must be chosen carefully to properly represent the physical scenario. Initial momentum was set to zero for all simulations in this report.

Momentum in `spiffe` is dimensionless, normalized to electron mass and the speed of light ($m_e c$). If the initial momentum is specified as 1×10^{-6} , this is equivalent to:

$$1 \times 10^{-6} m_e c = (1 \times 10^{-6})(9.109 \times 10^{-31} kg) \left(2.998 \frac{m}{s} \right) = 2.731 \times 10^{-36} \text{ kg-m/s} \quad (80)$$

f. Stiffness

`Spiffe` allows the user to alter the mass of the particle by adjusting beam stiffness. Stiffness refers to the particle mass in terms of electron mass. When simulating ions, the stiffness is increased to the appropriate value, but the charge will still be negative. The problem is simply set-up in reverse.

g. Current Density Profile

`Spiffe` provides an option to load a current density profile, allowing the user to gradually load a current density (e.g., a half-gaussian or linear ramp-up to the desired current density). This may be helpful for suppressing numerical oscillations while `spiffe` is converging to the solution. Use of this function requires caution when calculating the current density delivered. The screens that collect beam data must start after the ramp-up is complete. If the screens collect beam data during the ramp-up, then the calculated current density will appear low.

Minimal work was done with this function because the screens failed to start when specified. The current density was ramped linearly from $t=0$ to $t=1 \times 10^{-10}$ sec and then held constant from 1×10^{-10} sec to 5×10^{-10} sec. The screens were coded to start collecting data at 2×10^{-10} sec. The calculated current density from the screen was approximately 233 kA/m^2 , which happens to be the theoretical saturation current density. Closer examination revealed that the screen data included the linear ramp-up, so values from $J=0$ to $J=300 \text{ kA/m}^2$ were also included in the calculation. When the raw data was extracted from the simulation, the first data point in the first screen corresponded to 1×10^{-11} sec, when it should have been on or near 2×10^{-10} sec. After eliminating the undesirable points ($t < 2 \times 10^{-10}$ sec), the calculated current density was approximately 270 kA/m^2 .

h. Example Code

The code used for both the box and parallel plate geometries is shown here. Current density was varied manually for separate trials, although this can be done automatically with additional programming tools. `Number_per_step` and

`start_time` were constant for all trials. Time is given in seconds; positions are given in meters. This section of code should be added below the `define_geometry` section of the text file.

```
&define_cathode
    z_position=0,
    outer_radius=5e-3,
    current_density=300e3,
    start_time=1e-11,
    stop_time=2e-9,
    number_per_step=8,
    initial_pz=0,
&end
```

4. Poisson Correction

The field propagation routine uses two of Maxwell's equations: Ampere's law and Faraday's law. It does not use the divergence equation. Accumulated errors in the field propagation routine lead to non-zero divergence. `Spiffe` corrects for this by solving the Poisson equation at user-defined intervals during the simulation. Varying the options in this section of the code had minimal impact on the simulation results.

a. Correction Interval

The settings used in the Poisson correction section of the code have minimal impact on simulation time and current density output. The correction interval used throughout this report was 32. If the correction interval is increased to 320, the simulation finishes about two minutes faster. The current density delivered is similar - 270.7 kA/m² versus 271.3 kA/m² for 32 and 320, respectively.

b. Accuracy

Poisson correction accuracy is the fractional accuracy of the Poisson solutions. Its impact on the simulation is minimal. With the fractional accuracy set to 1×10^{-4} , `spiffe` returned a current density of 270.743 kA/m². Increasing the fractional accuracy by two orders of magnitude caused a negligible change in the output, 270.740 kA/m².

c. Iterations

The default number of iterations is 1000. If `spiffe` does not converge to a solution within the defined number of iterations, the program will provide a warning.

```
Warning: maximum number of iterations exceeded in
solve_poisson_cyl-residual was 3.952478e+001.
```

The simulations used throughout this report left the maximum number of iterations at the default value. The impact of this option on the program results is minimal. For comparison, if the number of iterations was limited to 100, `spiffe` provided approximately 80 warnings, each with a residual on the order of 10^8 . With 300 kA/m² requested, `spiffe` delivered 270.809 kA/m². With the default number of iterations, `spiffe` delivered 270.216 kA/m².

d. Example Code

The code below was used for all simulations in this report, regardless of geometry, run time, and current requested.

```
&poisson_correction
    step_interval=32,
    accuracy=1e-4,
    verbosity=0,
    error_charge_threshold=1e-15,
&end
```

5. Screens

The `define_screen` command is described in Section 4.13 of [14]. This option is necessary to determine current density output. With the simple geometries in this report, none of the screens observed backward particles. In more complicated geometries, it may be necessary to record both forward and backward particles. Screens may be placed individually using separate commands, or one command may be used to place several screens at regular intervals. Position and start time are in meters and seconds, respectively. The data is saved to the file that is specified by the user. In the example below, the file will be saved to *box.sc0*. Each `define_screen` command

should be given a different extension to avoid overwriting data (e.g., “.sc0”, “.sc1”, “.sc2”).

```
&define_screen
    filename="%s.sc0",
    z_position=1e-5,
    direction=forward,
&end
```

The screen output file (e.g., *box.sc0*) contains information on every macroparticle that passes through the screen.

- Radius (r) in meters; the radius of the particle when it crosses the screen.
- Momentum (pz, pr, pphi) in terms of $m_e c$, the momentum of the particle when it crosses the screen.
- Time (t) in seconds; time that the particle crosses the screen.
- Charge (q) in Coulombs; will be a multiple of electron charge, corresponding to the number of electrons per macroparticle.
- Initial radius (r0); radius that the particle was initially generated.
- Initial time (t0); time that the particle was initially generated.
- particleID; each macroparticle is assigned an ID number when it is generated.
- Minimum and maximum positions (rmin, rmax, zmin, zmax); minimum and maximum position of the particle from initial generation until it crosses the screen.

6. Snapshots

Beam snapshots are described in Section 4.15 of [14]. Beam snapshots are necessary for viewing particle trajectories. They are also helpful for troubleshooting unexpected results or problematic code.

In the following code, the beam data is saved to *box.snap*. The output file may be given any extension..

```
&define_snapshots
    filename="box.snap"
    time_interval=5e-11,
    start_time=0,
&end
```

The snapshot output file contains data on every macroparticle for each snapshot in time:

- Position (z , r) in meters.
- Momentum (p_z , p_r , p_{ϕ}) in terms of $m_e c$.
- Charge (q) in Coulombs; will be a multiple of electron charge, corresponding to the number of electrons per macroparticle.
- Initial radial position (r_0); the radius at which the particle was initially generated (random).
- Initial axial position (z_0); the longitude at which the particle was initially generated (should correspond to `z_position` from `define_cathode`).
- Initial time (t_0); time that the particle was initially generated.
- `particleID`; each macroparticle is assigned an ID number when it is generated.

7. Field Output Options

a. Field Output.

`Spiffe` provides two methods of examining the electric field. The first, `define_field_output`, maps the electric and magnetic field of the entire simulation area. This is described in Section 4.16 of [14]. The output file may be given any name and any extension. Individual data will rarely be pulled from this output file; field plots are more useful. Field plots are obtained using the `sddscontour` command.

b. Field Sampling

The second method of examining the field is through `define_field_sampling`. This command provides several options for examining a field component at a given position. The output file may be given any name and extension desired. Standard practice was to use an extension that described the component being sampled and the direction along which it was sampled. For example,

when sampling E_z along z at the centerline, the extension was “.Ez_z”. The option used throughout this report was to sample E_z along the centerline. This was accomplished with the following code:

```
&define_field_sampling
    filename="%s.Ez_z"
    component=Ez,
    min_coord=0,
    max_coord=0.0001 (0.000105)
    position=0,
    direction="z",
    time_interval=5e-11,
    start_time=0,
&end
```

When plotted, the E_z data points will correspond to the E_z grid rather than the user-defined grid. If the grid lines (defined by nz) are at $z=0$ and $z=5\text{ mm}$, then the first E_z grid point will lie at $z=2.5\text{ mm}$. In the case of the parallel plate geometry, max_coord was set to 0.105 mm to capture the final E_z grid point.

The field sampling command also provides an option for determining the average value of the field along a line. This option was not used for this report. This function did not work when requesting a time average along r .

8. Integration

a. Time Step

The simulation step size (dt_integration) is determined by the number of grid lines. If the step size is too large for the grid, `spiffe` will provide a warning.

A small time step will preclude longer simulation times due to memory limitations. The total number of steps is limited to 2^{31} steps (32-bit signed integer). If the time step is 1×10^{-14} seconds, then the simulation is limited to approximately $21.5\text{ }\mu\text{s}$.

$$231 \text{ steps} \times \frac{1 \times 10^{-14} \text{ s}}{\text{step}} = 2.15 \times 10^{-5} \text{ s} \quad (81)$$

Smaller step sizes will increase the run time. More gridlines require a smaller step size. There is a tradeoff between accuracy, memory, and time.

b. Finish Time

This should match `stop_time` as specified in the `define_cathode` section of the code.

c. Space Charge

This option allows the user to turn the space charge on and off. If zero, the Poisson correction becomes a Laplace correction. The fields generated by the simulation will match those learned in an introductory course in electrostatics.

With the space charge on, `spiffe` calculates the field from each macroparticle and its effect on the external field as well as neighboring particles. This requires considerably more computational steps and much longer run times.

d. Sample Code

The code used for both the box and parallel plate geometries is shown below. `Finish_time` always matched `stop_time` (from `define_cathode` section). Time is given in seconds. Section 4.19 of [14] provides a description of additional integration options and their default values. This section of code should be added at the bottom of the text file, after all the other code has been written. This is the only section of the code that is not optional. If this section is not included, `spiffe` will not calculate anything. Integration step size and run time were chosen using trial and error to find the best balance of accuracy and reasonable run times.

```
&integrate
    dt_integration=1e-14,
    start_time=0,
    finish_time=5e-10,
    space_charge=1,
&end
```

G. POST-PROCESSING

This section provides an introduction to the most commonly used post-processing tools from SDDS Toolkit. The SDDS Toolkit user's guide [15] provides more detailed information and options. The intent is to guide the user to useful commands, which can then be learned in more detail from [15].

1. **sddsquery**

This command is helpful for determining the type of data contained in any output file. Typing “sddsquery box.sc0” at the command prompt will return a list of the data name, units, symbol, and type contained in the file *box.sc0*. The data name is the terminology used to call on the data with other commands (e.g., sddsplot).

Data is considered either a parameter or a column. The simplest way to explain this is through an example. The file *box.sc0* was created from a `define_screen` command. This command captures beam data (r , p_z , p_r , t , q , etc.) at each time step. The output file is separated into tables; each table represents data from one time step. The tables are labeled with the time step of the data they contain; time step is the parameter. The beam data contained in the table is separated into columns.

2. **sdds2spreadsheet**

This command has been replaced by `sddsprintout`, but it is helpful for understanding the layout of an output file. To view *box.sc0* as a text file (*box.txt*), type the following at the command prompt:

```
sdds2spreadsheet box.sc0 box.txt -delim="|"
```

The delimiter option (`-delim="|"`) is entirely optional. It is helpful for importing data into other programs.

3. **sddscombine**

The output files are separated into tables, each with a set of parameters as the header. The user's guide [15] describes using this command to combine several files into

a single file. Another option is to use it to combine the separate tables in one file into a single table. This is helpful because many programs will only see the first table during a data import. This command was the first step in calculating current through a screen. The following command combines the separate tables of *box.sc0* into a single table in the file *box.comb*.

```
sddscombine box.sc0 box.comb -merge -overwrite
```

4. sddsprocess

Spiffe data can be exported to a spreadsheet or Matlab[®] for processing, but the `sddsprocess` command performs many of the same functions. This command was used to calculate the current density through each screen. First, a screen was placed in the beam path using the `define_screen` command in the run file. Next, the data from the screen file (*box.sc0*) was combined into the file *box.comb*. This file contains several columns of beam data, but the relevant columns for calculating current are time (*t*) and charge (*q*). Each time a particle passes through the screen, its charge and the time it passes through the screen are recorded. Calculating the current simply requires knowledge of the total charge that passed through the screen in a given amount of time. The current density is calculated by dividing by the area of the beam. This is all accomplished with one `sddsprocess` command. The output is sent to the file *box.pro*.

```
sddsprocess box.comb box.pro -process=q,sum,qsum
-process=t,spread,tspread -define=parameter,I,"qsum tsread
/",units=A -define=parameter,J,"I 5e-3 sqr pi *
/",units=A/m$a2.
```

The notation “m\$a2” equates to m^2 . The notation “\$a” is telling the program to place the next character in the superscript position. The subscript position is accomplished using “\$b”. The text is returned to the normal position using “\$n”.

Additional parameters may be defined by adding more `-process` options to the line above. To calculate the average longitudinal momentum (*pz*), add “`-process=pz,average,pravg`” to the line above. Average is a built-in process

for `sddsprocess`, `pz` is the name of the column data that is being processed, and `pravg` is the name of the new parameter as defined by the user.

5. **sddscollapse**

The current density (J) that was calculated using `sddsprocess` was defined as a parameter. If several parameters are defined (e.g, `rmax`, `pxmin`, `pzmax`, `pzavg`), it is helpful to use the `sddscollapse` command. This command will collapse all the parameters from the original file (e.g., *box.pro*) into a set of columns. The column data from the original file will not be transferred. The columns of the new file (*box.coll*) will be created from the parameters of the original file.

```
sddscollapse box.pro box.coll
```

6. **sddsprintout**

The data contained in any sdds file can be viewed using `sddsprintout`. The user must specify which data to print; this can be columns, parameters, or both. To view the current density (J) and average longitudinal momentum (`pzavg`) from *box.coll*, type the following command at the command prompt.

```
sddsprintout box.coll box.txt -col=(J,pzavg)
```

The data is printed to the file *box.txt*, which the user must open to view the required data.

7. **sddscontour**

This command was used to create the electric field contour plots throughout this report. This is the simplest plotting tool available in SDDS Toolkit. A basic contour plot of E_z can be obtained using the following command:

```
sddcontour -quantity=Ez box.fields
```

The number of contour lines may be adjusted by adding the “-contours” switch:

```
sddscontour -quantity=Ez -contours=50 box.fields
```


The contour plot may be shaded by adding the “-shade” switch:

```
sddscontour -quantity=Ez -shade box.fields
```

8. **sddsplot**

This command has a multitude of options available for the user. It can create quick plots for viewing on-screen or it can create publication-quality graphics. One of the most useful ways to utilize this command is to view beam snapshots. First, snapshots must be added to the run program using `define_snapshots`. The plotting command must call on the particle location (z, r) from the snapshot file (*box.snap*) as well as the coordinates in the boundary file (*box.dbnd*). To obtain an on-screen plot, type the following command at the prompt:

```
sddsplot -col=z,r -graph=dot -split=page -sep box.snap  
-col=z,r -graph=line -split=page -sep -omni box.dbnd
```

Snapshots are viewed by pressing “n” for next and “p” for previous. If `-sep` is not used, the snapshots will be plotted on top of each other rather than in separate windows. This is helpful for showing the beam envelope for the entire simulation.

A publication-quality graphic is obtained by requesting a white background and sending the plot to an output file (*snap.png*).

```
sddsplot -col=z,r -graph=dot -split=page -sep box.snap  
-col=z,r -graph=line -split=page -sep -omni box.dbnd
```

This command will only plot the first snapshot (first table of data). To plot the remaining snapshots, the snapshot file (*box.snap*) must be split into separate files using the `sddssplit` command.

9. **sddssplit**

Output files contain several tables of data, with each table corresponding to a different time step. The `sddssplit` command creates several new files from the

original file. Each new file contains data from one table of the original file. To split *box.snap* into several individual snapshot files, type the following command at the prompt:

```
sddssplit box.snap -rootname=box -digits=3 -  
extension=snap
```

If the original file contains 11 tables of data, this command will create 11 files named *box001.snap*, *box002.snap*, ..., *box011.snap*. Each of these snapshot files may then be plotted to an output file using the `sddsplo`t command described previously.

H. TROUBLESHOOTING

1. General Troubleshooting

The boundary output should always be compared before proceeding with any simulations or troubleshooting. This was described in Section F.2.c of this Appendix.

One of the easiest methods for verifying a working code is by viewing beam snapshots. If there are no particles in any of the snapshots, something is probably wrong with the code. The most likely causes are:

- Voltage is not high enough to accelerate the particles in the given amount of simulation time. This is fixed by increasing either the voltage or simulation time.
- Cathode `stop_time` does not match simulation `finish_time`. This only seemed to be a problem on Windows-based machines.
- Cathode location might be inside a boundary. Make sure that the discrete boundary matches the ideal boundary and verify the location of the cathode.
- Wrong dimensions. Verify all dimensions are in meters, including cavity boundaries and cathode location.

A second method of checking for particles is by viewing screen data. Use `sddsprintout` or `sdds2spreadsheet` to view the data. If the file is empty, no particles have passed through the screen in the direction indicated. It may be helpful to add another screen at the same location to record backward particles.

Viewing the electric field contours is another way to verify that the code is working as expected. Run a simulation without particles to verify that the electric field is established as expected. Running the simulation with the space charge off is another option to check the electric fields. If the electric fields are not established as expected, verify the potentials and dimensions of the geometry file.

2. Files not Found

Reference [14] shows the use of both commas and semi-colons following a command/namelist. Use caution when using semi-colons. For example, using

```
boundary_output=box.bnd,
```

creates an output file named *box.geo*, but using

```
boundary_output=box.geo;
```

creates an output file name *box.geo*; (note that the semi-colon is now a part of the file name).

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX B SCREEN DATA

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
5	0	4.991512	6.259E-03	0	4.999995
	1	4.990803	1.399E-02	0	4.999995
	2	4.991676	1.877E-02	0	4.999995
50	0	49.91512	6.259E-03	0	4.999995
	1	49.90803	1.399E-03	0	4.999995
	2	49.91676	1.877E-02	0	4.999995
100	0	99.83024	6.259E-03	0	4.999995
	1	99.81605	1.399E-03	0	4.999995
	2	99.83352	1.877E-02	0	4.999995
150	0	149.7454	6.259E-03	0	4.999995
	1	149.7241	1.399E-02	0	4.999995
	2	149.7503	1.877E-03	0	4.999995
200	0	199.6605	6.259E-03	0	4.999995
	1	199.6321	1.399E-02	0	4.999995
	2	199.667	1.877E-02	0	4.999995
250	0	249.5756	6.259E-03	0	4.999995
	1	249.5401	1.399E-02	0	4.999995
	2	249.5838	1.877E-02	0	4.999995
300	0	299.4907	6.259E-03	0	4.999995
	1	299.4482	1.399E-02	0	4.999995
	2	299.5006	1.877E-02	0	4.999995
350	0	349.4058	6.259E-03	0	4.999995
	1	349.3562	1.399E-02	0	4.999995
	2	349.4173	1.877E-02	0	4.999995
400	0	399.3209	6.259E-03	0	4.999995
	1	399.2642	1.399E-02	0	4.999995
	2	399.3341	1.877E-02	0	4.999995
450	0	449.2361	6.259E-03	0	4.999995
	1	449.1722	1.399E-02	0	4.999995
	2	449.2508	1.877E-02	0	4.999995
500	0	499.1512	6.259E-03	0	4.999995
	1	499.0803	1.399E-02	0	4.999995
	2	499.1676	1.877E-02	0	4.999995

Table 27. Screen data, box, space charge off.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
5	0	4.990588	6.229E-03	-1.658E-08	5.000001
	1	4.989335	1.396E-02	-9.347E-09	5.000076
	2	4.990293	1.876E-02	-1.141E-07	5.000168
50	0	49.83783	5.976E-03	-1.442E-07	5.000058
	1	49.78213	1.372E-02	1.204E-09	5.000835
	2	49.78434	1.871E-02	1.028E-06	5.001752
100	0	99.51599	5.661E-03	-2.198E-07	5.000131
	1	99.31531	1.344E-02	3.697E-07	5.001751
	2	99.28589	1.864E-02	-1.549E-06	5.003623
150	0	148.9618	5.293E-03	-1.830E-07	5.000217
	1	148.4779	1.313E-02	1.378E-06	5.002786
	2	148.3694	1.857E-02	-1.314E-06	5.005690
200	0	197.9707	4.840E-03	1.166E-07	5.000320
	1	196.9827	1.278E-02	3.378E-06	5.004042
	2	196.7009	1.850E-02	1.782E-07	5.008131
250	0	245.6037	4.211E-03	9.568E-07	5.000465
	1	243.7504	1.239E-02	7.391E-06	5.005637
	2	243.2093	1.842E-02	4.033E-06	5.011119
300	0	270.216	3.484E-03	3.942E-06	5.000716
	1	267.0417	1.211E-02	1.445E-05	5.007847
	2	266.0374	1.838E-02	1.161E-05	5.015166
350	0	271.2137	3.385E-03	3.364E-06	5.001017
	1	268.0365	1.209E-02	1.595E-05	5.010126
	2	266.778	1.838E-02	1.373E-05	5.018893
400	0	270.7643	3.343E-03	4.043E-06	5.001373
	1	267.4595	1.208E-02	1.700E-05	5.011907
	2	266.4258	1.838E-02	1.453E-05	5.021549
450	0	271.6229	3.326E-03	5.416E-06	5.001679
	1	267.3812	1.208E-02	1.837E-05	5.012804
	2	267.0899	1.838E-02	1.565E-05	5.022751
500	0	271.5669	3.314E-03	4.669E-06	5.002326
	1	268.3792	1.208E-02	1.869E-05	5.015046
	2	266.9832	1.838E-02	1.628E-05	5.025171

Table 28. Screen data, box, space charge on.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
5	0	4.991512	6.256E-03	0	4.999995
	1	4.990803	1.399E-02	0	4.999995
	2	4.991676	1.877E-02	0	4.999995
50	0	49.91512	6.256E-03	0	4.999995
	1	49.90803	1.399E-02	0	4.999995
	2	49.91676	1.877E-02	0	4.999995
100	0	99.83024	6.256E-03	0	4.999995
	1	99.81605	1.399E-02	0	4.999995
	2	99.83352	1.877E-02	0	4.999995
150	0	149.7454	6.256E-03	0	4.999995
	1	149.7241	1.399E-02	0	4.999995
	2	149.7503	1.877E-02	0	4.999995
200	0	199.6605	6.256E-03	0	4.999995
	1	199.6321	1.399E-02	0	4.999995
	2	199.667	1.877E-02	0	4.999995
250	0	249.5756	6.256E-03	0	4.999995
	1	249.5401	1.399E-02	0	4.999995
	2	249.5838	1.877E-02	0	4.999995
300	0	299.4907	6.256E-03	0	4.999995
	1	299.4482	1.399E-02	0	4.999995
	2	299.5006	1.877E-02	0	4.999995
350	0	349.4058	6.256E-03	0	4.999995
	1	349.3562	1.399E-02	0	4.999995
	2	349.4173	1.877E-02	0	4.999995
400	0	399.3209	6.256E-03	0	4.999995
	1	399.2642	1.399E-02	0	4.999995
	2	399.3341	1.877E-02	0	4.999995
450	0	449.2361	6.256E-03	0	4.999995
	1	449.1722	1.399E-02	0	4.999995
	2	449.2508	1.877E-02	0	4.999995
500	0	499.1512	6.256E-03	0	4.999995
	1	499.0803	1.399E-02	0	4.999995
	2	499.1676	1.877E-02	0	4.999995

Table 29. Screen data, parallel plate, space charge off.

J_{req} (kA/m ²)	Screen	J_{del} (kA/m ²)	$p_{z,\text{avg}}$ (m _e c)	$p_{r,\text{avg}}$ (m _e c)	r_{max} (mm)
5	0	4.990464	6.213E-03	-1.186E-08	5.000000
	1	4.988699	1.393E-02	2.341E-10	5.000071
	2	4.989187	1.871E-02	-9.912E-08	5.000159
50	0	49.7854	5.793E-03	-6.424E-08	5.000056
	1	49.64779	1.333E-02	3.014E-07	5.000874
	2	49.57408	1.819E-02	-5.570E-07	5.001852
100	0	99.08017	5.239E-03	6.545E-08	5.000176
	1	98.39032	1.261E-02	1.659E-06	5.002332
	2	98.02934	1.760E-02	3.218E-07	5.004700
150	0	146.5805	4.517E-03	6.389E-07	5.000421
	1	144.4719	1.182E-02	5.052E-06	5.005229
	2	143.7236	1.702E-02	3.715E-06	5.009518
200	0	176.3814	3.712E-03	2.529E-06	5.001135
	1	174.034	1.118E-02	1.137E-05	5.009485
	2	174.101	1.660E-02	1.071E-05	5.017878
250	0	184.8474	3.284E-03	4.302E-06	5.001837
	1	183.1321	1.091E-02	1.608E-05	5.013340
	2	183.5346	1.645E-02	1.582E-05	5.023391
300	0	187.3599	3.112E-03	4.479E-06	5.002438
	1	184.9992	1.083E-02	1.831E-05	5.015919
	2	185.8662	1.641E-02	1.830E-05	5.027015
350	0	188.0085	3.050E-03	4.414E-06	5.003105
	1	185.6726	1.080E-02	1.957E-05	5.017951
	2	186.3659	1.640E-02	1.974E-05	5.029710
400	0	187.8295	3.028E-03	9.802E-06	5.003967
	1	185.6414	1.080E-02	2.188E-05	5.020319
	2	186.3309	1.640E-02	2.123E-05	5.032293
450	0	187.9653	3.013E-03	8.929E-06	5.004827
	1	185.9714	1.079E-02	2.201E-05	5.020169
	2	186.7722	1.639E-02	2.185E-05	5.032424
500	0	187.8994	2.996E-03	6.467E-06	5.006683
	1	186.4457	1.077E-02	2.156E-05	5.024437
	2	187.3612	1.638E-02	2.210E-05	5.037074

Table 30. Screen data, parallel plate, space charge on.

LIST OF REFERENCES

- [1] R. G. Jahn, *Physics of Electric Propulsion*. New York: McGraw-Hill, 1968.
- [2] E. Stuhlinger, *Ion Propulsion for Space Flight*. New York: McGraw-Hill, 1964.
- [3] D. M. Goebel and I. Katz, *Fundamentals of Electric Propulsion: Ion and Hall Thrusters*. Hoboken, N.J.: Wiley, 2008.
- [4] G. Sutton and O. Biblarz, *Rocket Propulsion Elements*. Hoboken, N.J.: Wiley, 2010.
- [5] G. R. Brewer, *Ion Propulsion, Technology and Applications*. New York: Gordon and Breach, 1970.
- [6] R. Gomer, *Field Emission and Field Ionization*. American Institute of Physics, New York, 1962.
- [7] B. Shrank, “Development and testing of a field ionized ion thruster for microsatellite applications,” M.S. thesis, Naval Postgraduate School, Monterey, CA, 2010.
- [8] P. Camp, “Design and installation of a field ionization test chamber for ion thrusters,” M.S. thesis, Naval Postgraduate School, Monterey, CA, 2010.
- [9] J. R. Harris and J. W. Lewellen, “Transmission of intense electron beams through apertures,” *Physics of Plasmas*, vol 17, April 2010.
- [10] J. R. Harris and J. W. Lewellen, “Suppression of current fluctuations in an intense electron beam,” *Journal of Applied Physics*, vol. 108, October 2010.
- [11] D. Tskhakaya, “Chapter 6: The Particle-in-Cell Method,” in *Computational Many-Particle Physics*, Lecture Notes in Physics, vol. 739, 2008.
- [12] M. Borland, “Summary of Equations and Methods Used in spiffe,” *Advanced Photon Source*, June 29, 1992, APS/IN/LINAC/92-2.
- [13] J. D. Jackson, *Classical Electrodynamics*. Wiley, New York, 1998.
- [14] M. Borland, “User’s Guide for spiffe Version 3.2,” Advanced Photon Source, June 28, 2005, http://www.aps.anl.gov/Accelerator_Systems_Division/Accelerator_Operations_Physics/manuals/spiffe_V3.2/spiffe.html.

- [15] M. Borland, L. Emery, H. Shang, and R. Soliday, “User’s Guide for SDDS Toolkit Version 2.8,” Advanced Photon Source, June 22, 2010, http://www.aps.anl.gov/Accelerator_Systems_Division/Accelerator_Operations_Physics/manuals/SDDStoolkit/SDDStoolkit.html.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
Ft. Belvoir, Virginia
2. Dudley Knox Library
Naval Postgraduate School
Monterey, California
3. Professor Oscar Biblarz
Department of Mechanical and Aerospace Engineering
Naval Postgraduate School
Monterey, California
4. Dragoslav Grbovic
Department of Physics
Naval Postgraduate School
Monterey, California
5. Professor John Lewellen
Department of Physics
Naval Postgraduate School
Monterey, California
6. Dr. Michael Borland
Accelerator Systems Division
Argonne National Laboratory
Argonne, Illinois